

(11)特許出願公開番号

(43)公開日 平成11年(1999)1月29日

H0 4 N 1/419

(74)代理人 弁理士 佐藤 幸男 (外1名)

● 説明図 紙の造り方と化学物質の製造

**【特許請求の範囲】**

**【請求項1】** 2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼び、入力された前記2値画像または前記2値画像のランレングス符号に基づいて、前記2値画像の主走査線毎に、主走査線内のラン数およびラン情報を解析するラン解析工程と、

前回、前記ラン解析工程により解析された直前の主走査線のラン数と、今回、前記ラン解析工程により解析された主走査線のラン数とを比較する比較工程と、前記ラン解析工程により解析された主走査線のラン数が1回以上連続して0である回数を表すオフセット値をカウントするカウント工程と、

後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが主走査線のラン数である旨を表す符号データをラン命令と呼ぶとき、

今回、前記ラン解析工程により解析された主走査線のラン数が0でない場合、

前回、前記ラン解析工程により解析された直前の主走査線のラン数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、

次に、前記比較工程の比較結果が不一致のとき、前記ラン命令と、前記ラン解析工程により解析されたラン数と、前記ラン解析工程により解析されたラン情報とを結合して符号を構成し、

前記比較工程の比較結果が一致のとき、前記ラン解析工程により解析されたラン情報により符号を構成する符号構成工程とを含むことを特徴とする画像符号化方法。

**【請求項2】** 符号化された2値画像の符号データを複数の画素からなる2値画像に復号化する画像復号化方法であって、

前記2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、

前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼び、

全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、

後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記ラン数である旨を表す符号データをラン命令と呼ぶとき、

入力された符号データに含まれるオフセット命令および

ラン命令を解析する命令解析工程と、

前記命令解析工程によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、前記命令解析工程によりラン命令が解析された場合、ラン命令に後続するラン数を解析し、解析されたラン数に後続するラン情報を前記ラン数毎に取り出して主走査線を復号化する復号化工程とを含むことを特徴とする画像復号化方法。

**【請求項3】** 請求項1記載の画像符号化方法と請求項2記載の画像復号化方法とを有することを特徴とする画像符号化復号化方法。

**【請求項4】** 2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼び、

入力された前記2値画像または前記2値画像のランレングス符号に基づいて、前記2値画像の主走査線毎に、主走査線内のラン数およびラン情報を解析するラン解析部と、

前回、前記ラン解析部により解析された直前の主走査線のラン数と、今回、前記ラン解析部により解析された主走査線のラン数とを比較する比較部と、

前記ラン解析部により解析された主走査線のラン数が1回以上連続して0である回数を表すオフセット値をカウントするカウンタと、

後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが主走査線のラン数である旨を表す符号データをラン命令と呼ぶとき、

今回、前記ラン解析部により解析された主走査線のラン数が0でない場合、

前回、前記ラン解析部により解析された直前の主走査線のラン数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、

次に、前記比較部の比較結果が不一致のとき、前記ラン命令と、前記ラン解析部により解析されたラン数と、前記ラン解析部により解析されたラン情報とを結合して符号を構成し、

前記比較部の比較結果が一致のとき、前記ラン解析部により解析されたラン情報により符号を構成する符号構成部とを備えたことを特徴とする画像符号化装置。

**【請求項5】** 符号化された2値画像の符号データを複数の画素からなる2値画像に復号化する画像復号化装置であって、

前記2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、

前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点およ

び終点、または、始点および画素数を表す情報をラン情報と呼び、

全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、

後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記ラン数である旨を表す符号データをラン命令と呼ぶとき、

入力された符号データに含まれるオフセット命令およびラン命令を解析する命令解析部と、

前記命令解析部によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、前記命令解析部によりラン命令が解析された場合、ラン命令に後続するラン数を解析し、解析されたラン数に後続するラン情報を前記ラン数毎に取り出して主走査線を復号化する復号化部とを備えたことを特徴とする画像復号化装置。

【請求項6】 請求項4記載の画像符号化装置と請求項5記載の画像復号化装置とを有することを特徴とする画像符号化復号化装置。

【請求項7】 2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、

前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼ぶとき、

入力された前記2値画像または前記2値画像のランレングス符号に基づいて、前記2値画像の主走査線毎に、主走査線内のラン数およびラン情報を解析するラン解析工程と、

前回、前記ラン解析工程により解析された直前の主走査線のラン数と、今回、前記ラン解析工程により解析された主走査線のラン数との差を算出する算出工程と、

前記ラン解析工程により解析された主走査線のラン数が1回以上連続して0である回数を表すオフセット値をカウントするカウント工程と、

後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、符号化すべき主走査線のラン数が直前の主走査線のラン数より多い旨を表す符号データを進入命令と呼び、符号化すべき主走査線のラン数が直前の主走査線のラン数より少ない旨を表す符号データを退出命令と呼ぶとき、

今回、前記ラン解析工程により解析された主走査線のラン数が0でない場合、

前回、前記ラン解析工程により解析された直前の主走査線のラン数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、

次に、前記算出工程の算出結果が符号化すべき主走査線のラン数が直前の主走査線のラン数より多い旨を意味す

るとき、前記進入命令と、前記算出工程の算出結果の絶対値と、前記ラン解析工程により解析されたラン情報とを結合して符号を構成し、

前記算出工程の算出結果が符号化すべき主走査線のラン数が直前の主走査線のラン数より少ない旨を意味するとき、前記退出命令と、前記算出工程の算出結果の絶対値と、前記ラン解析工程により解析されたラン情報とを結合して符号を構成し、

前記算出工程の算出結果が符号化すべき主走査線のラン数と直前の主走査線のラン数とが同数である旨を意味するとき、前記ラン解析工程により解析されたラン情報により符号を構成する符号構成工程とを含むことを特徴とする画像符号化方法。

【請求項8】 符号化された2値画像の符号データを複数の画素からなる2値画像に復号化する画像復号化方法であって、

前記2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、

前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼び、

全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、

後続する符号データがオフセット値である旨を表す符号データをオフセット命令と呼び、復号化すべき主走査線のラン数が直前の主走査線より多い旨を表す符号データを進入命令と呼び、復号化すべき主走査線のラン数が直前の主走査線より少ない旨を表す符号データを退出命令と呼び、

復号化すべき主走査線のラン数と直前の主走査線のラン数との差の絶対値を差分絶対値と呼ぶとき、

入力された符号データに含まれるオフセット命令、進入命令および退出命令を解析する命令解析工程と、

既に復号化済みの直前の主走査線のラン数を保持し、前記命令解析工程により進入命令が解析された場合、進入命令に後続する前記差分絶対値を、直前の主走査線のラン数に加算し、前記命令解析工程により退出命令が解析された場合、退出命令に後続する前記差分絶対値を、直前の主走査線のラン数から減算し、復号化すべき主走査線に含まれるラン数を算出する算出工程と、

前記命令解析工程によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、

前記命令解析工程により進入命令が解析された場合および前記命令解析工程により退出命令が解析された場合、前記差分絶対値に後続するラン情報を、前記算出工程により算出されたラン数毎に取り出して主走査線を復号化する復号化工程とを含むことを特徴とする画像復号化方

法。

【請求項9】 請求項7記載の画像符号化方法と請求項8記載の画像復号化方法とを有することを特徴とする画像符号化復号化方法。

【請求項10】 2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼ぶとき、  
入力された前記2値画像または前記2値画像のランレングス符号に基づいて、前記2値画像の主走査線毎に、主走査線内のラン数およびラン情報を解析するラン解析部と、  
前回、前記ラン解析部により解析された直前の主走査線のラン数と、今回、前記ラン解析部により解析された主走査線のラン数との差を算出する算出部と、  
前記ラン解析部により解析された主走査線のラン数が1回以上連続して0である回数を表すオフセット値をカウントするカウンタと、  
後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、符号化すべき主走査線のラン数が直前の主走査線のラン数より多い旨を表す符号データを進入命令と呼び、符号化すべき主走査線のラン数が直前の主走査線のラン数より少ない旨を表す符号データを退出命令と呼ぶとき、  
今回、前記ラン解析部により解析された主走査線のラン数が0の場合、  
前回、前記ラン解析部により解析された直前の主走査線のラン数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、  
次に、前記算出部の算出結果が符号化すべき主走査線のラン数が直前の主走査線のラン数より多い旨を意味するとき、前記進入命令と、前記算出部の算出結果の絶対値と、前記ラン解析部により解析されたラン情報とを結合して符号を構成し、  
前記算出部の算出結果が符号化すべき主走査線のラン数が直前の主走査線のラン数より少ない旨を意味するとき、前記退出命令と、前記算出部の算出結果の絶対値と、前記ラン解析部により解析されたラン情報とを結合して符号を構成し、  
前記算出部の算出結果が符号化すべき主走査線のラン数と直前の主走査線のラン数とが同数である旨を意味するとき、前記ラン解析部により解析されたラン情報により符号を構成する符号構成部とを備えたことを特徴とする画像符号化装置。  
【請求項11】 符号化された2値画像の符号データを複数の画素からなる2値画像に復号化する画像復号化装置であって、  
前記2値画像が、図形を表す複数の黒画素と、図形の背

景を表す複数の白画素とを含み、

前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼び、

全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、

後続する符号データがオフセット値である旨を表す情報をオフセット命令と呼び、復号化すべき主走査線のラン数が直前の主走査線より多い旨を表す情報を進入命令と呼び、復号化すべき主走査線のラン数が直前の主走査線より少ない旨を表す情報を退出命令と呼び、

復号化すべき主走査線のラン数と直前の主走査線のラン数との差の絶対値を差分絶対値と呼ぶとき、

入力された符号データに含まれるオフセット命令、進入命令および退出命令を解析する命令解析部と、

既に復号化済みの直前の主走査線のラン数を保持し、前記命令解析部により進入命令が解析された場合、進入命令に後続する前記差分絶対値を、直前の主走査線のラン数に加算し、前記命令解析部により退出命令が解析された場合、退出命令に後続する前記差分絶対値を、直前の主走査線のラン数から減算して復号化すべき主走査線に含まれるラン数を算出する算出部と、

前記命令解析部によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、

前記命令解析部により進入命令が解析された場合および前記命令解析部により退出命令が解析された場合、前記差分絶対値に後続するラン情報を、前記算出部により算出されたラン数毎に取り出して主走査線を復号化する復号化部とを備えたことを特徴とする画像復号化装置。

【請求項12】 請求項10記載の画像符号化装置と請求項11記載の画像復号化装置とを有することを特徴とする画像符号化復号化装置。

【請求項13】 多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、

前記多値画像の主走査線内に1つ以上連続する第1画素の画素列を第1ランと呼び、前記多値画像の主走査線内に1つ以上連続する第2画素の画素列を第2ランと呼び、

前記主走査線内の第1ランの始点、終点および第1画素値、または、始点、画素数および第1画素値を表す情報を第1ラン情報と呼び、前記主走査線内の第2ランの始点、終点および第2画素値、または、始点、画素数および第2画素値を表す情報を第2ラン情報と呼び、

入力された前記多値画像に基づいて、前記多値画像の主

走査線毎に、主走査線内の第1および第2ランの総数、並びに、第1および第2ラン情報を解析するラン解析工程と、

前回、前記ラン解析工程により解析された直前の主走査線の第1および第2ランの総数と、今回、前記ラン解析工程により解析された主走査線の第1および第2ランの総数とを比較する比較工程と、

前記ラン解析工程により解析された主走査線の第1および第2ランの総数が1回以上連続して0である回数を表すオフセット値をカウントするカウント工程と、

後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記第1および第2ランの総数である旨を表す符号データをラン命令と呼ぶとき、

今回、前記ラン解析工程により解析された主走査線の第1および第2ランの総数が0でない場合、

前回、前記ラン解析工程により解析された直前の主走査線の第1および第2ランの総数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、

次に、前記比較工程の比較結果が不一致のとき、前記ラン命令と、前記ラン解析工程により解析された第1および第2ランの総数と、前記ラン解析工程により解析された第1および第2ラン情報とを結合して符号を構成し、前記比較工程の比較結果が一致のとき、前記ラン解析工程により解析された第1および第2ラン情報により符号を構成する符号構成工程とを含むことを特徴とする画像符号化方法。

【請求項14】 符号化された多値画像の符号データを複数の画素からなる多値画像に復号化する画像復号化方法であって、

前記多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、

前記多値画像の主走査線内に1つ以上連続する第1画素の画素列を第1ランと呼び、前記多値画像の主走査線内に1つ以上連続する第2画素の画素列を第2ランと呼び、

前記主走査線内の第1ランの始点、終点および第1画素値、または、始点、画素数および第1画素値を表す情報を第1ラン情報と呼び、前記主走査線内の第2ランの始点、終点および第2画素値、または、始点、画素数および第2画素値を表す情報を第2ラン情報と呼び、

全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、

後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号デー

タが前記第1および第2ランの総数である旨を表す符号データをラン命令と呼ぶとき、

入力された符号データに含まれるオフセット命令およびラン命令を解析する命令解析工程と、

前記命令解析工程によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、前記命令解析工程によりラン命令が解析された場合、ラン命令に後続する第1および第2ランの総数を解析し、解析された第1および第2ランの総数に後続する第1および第2ラン情報を前記第1および第2ランの総数毎に取り出して主走査線を復号化する復号化工程とを含むことを特徴とする画像復号化方法。

【請求項15】 請求項13記載の画像符号化方法と請求項14記載の画像復号化方法とを有することを特徴とする画像符号化復号化方法。

【請求項16】 多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、

前記多値画像の主走査線内に1つ以上連続する第1画素の画素列を第1ランと呼び、前記多値画像の主走査線内に1つ以上連続する第2画素の画素列を第2ランと呼び、

前記主走査線内の第1ランの始点、終点および第1画素値、または、始点、画素数および第1画素値を表す情報を第1ラン情報と呼び、前記主走査線内の第2ランの始点、終点および第2画素値、または、始点、画素数および第2画素値を表す情報を第2ラン情報と呼び、

入力された前記多値画像に基づいて、前記多値画像の主走査線毎に、主走査線内の第1および第2ランの総数、並びに、第1および第2情報を解析するラン解析部と、前回、前記ラン解析部により解析された直前の主走査線の第1および第2ランの総数と、今回、前記ラン解析部により解析された主走査線の第1および第2ランの総数とを比較する比較部と、

前記ラン解析部により解析された主走査線の第1および第2ランの総数が1回以上連続して0である回数を表すオフセット値をカウントするカウンタと、

後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記第1および第2ランの総数である旨を表す符号データをラン命令と呼ぶとき、

今回、前記ラン解析部により解析された主走査線の第1および第2ランの総数が0でない場合、

前回、前記ラン解析部により解析された直前の主走査線の第1および第2ランの総数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、

次に、前記比較部の比較結果が不一致のとき、前記ラン命令と、前記ラン解析部により解析された第1および第

2ランの総数と、前記ラン解析部により解析された第1および第2ラン情報とを結合して符号を構成し、前記比較部の比較結果が一致のとき、前記ラン解析部により解析された第1および第2ラン情報により符号を構成する符号構成部とを備えたことを特徴とする画像符号化装置。

【請求項17】 符号化された多値画像の符号データを複数の画素からなる多値画像に復号化する画像復号化装置であって、

前記多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、

前記多値画像の主走査線内に1つ以上連続する第1画素の画素列を第1ランと呼び、前記多値画像の主走査線内に1つ以上連続する第2画素の画素列を第2ランと呼び、

前記主走査線内の第1ランの始点、終点および第1画素値、または、始点、画素数および第1画素値を表す情報を第1ラン情報と呼び、前記主走査線内の第2ランの始点、終点および第2画素値、または、始点、画素数および第2画素値を表す情報を第2ラン情報と呼び、全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、

後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記第1および第2ランの総数である旨を表す符号データをラン命令と呼ぶとき、

入力された符号データに含まれるオフセット命令およびラン命令を解析する命令解析部と、

前記命令解析部によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、前記命令解析部によりラン命令が解析された場合、ラン命令に後続する第1および第2ランの総数を解析し、解析された第1および第2ランの総数に後続する第1および第2ラン情報を前記第1および第2ランの総数毎に取り出して主走査線を復号化する復号化部とを備えたことを特徴とする画像復号化装置。

【請求項18】 請求項16記載の画像符号化装置と請求項17記載の画像復号化装置とを有することを特徴とする画像符号化復号化装置。

【請求項19】 多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、

前記多値画像の主走査線内に1つ以上連続する前記白画素以外の画素の画素列をランと呼び、前記主走査線内の

ランの始点およびその画素値、並びに、終点およびその画素値をラン情報と呼ぶとき、

入力された前記多値画像に基づいて、前記多値画像の主走査線毎に、主走査線内のラン数およびラン情報を解析するラン解析工程と、

前回、前記ラン解析工程により解析された直前の主走査線のラン数と、今回、前記ラン解析工程により解析された主走査線のラン数とを比較する比較工程と、

前記ラン解析工程により解析された主走査線のラン数が1回以上連続して0である回数を表すオフセット値をカウントするカウント工程と、

後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記ラン数である旨を表す符号データをラン命令と呼ぶとき、

今回、前記ラン解析工程により解析された主走査線のラン数が0でない場合、

前回、前記ラン解析工程により解析された直前の主走査線のラン数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、

次に、前記比較工程の比較結果が不一致のとき、前記ラン命令と、前記ラン解析工程により解析されたラン数と、前記ラン解析工程により解析されたラン情報とを結合して符号を構成し、

前記比較工程の比較結果が一致のとき、前記ラン解析工程により解析されたラン情報により符号を構成する符号構成工程とを含むことを特徴とする画像符号化方法。

【請求項20】 符号化された多値画像の符号データを複数の画素からなる多値画像に復号化する画像復号化方法であって、

前記多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、

前記多値画像の主走査線内に1つ以上連続する前記白画素以外の画素の画素列をランと呼び、前記主走査線内のランの始点およびその画素値、並びに、終点およびその画素値をラン情報と呼び、

全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、

後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記ラン数である旨を表す符号データをラン命令と呼ぶとき、

入力された符号データに含まれるオフセット命令およびラン命令を解析する命令解析工程と、

前記命令解析工程によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する

白ラインを復号化し、前記命令解析工程によりラン命令が解析された場合、ラン命令に後続するラン数を解析し、解析されたラン数に後続するラン情報を前記ラン数毎に取り出し、取り出されたラン情報に基づいて、ランの始点および終点間の画素の画素値を補間して主走査線を復号化する復号化工程とを含むことを特徴とする画像復号化方法。

【請求項21】 請求項19記載の画像符号化方法と請求項20記載の画像復号化方法とを有することを特徴とする画像符号化復号化方法。

【請求項22】 多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、前記多値画像の主走査線内に1つ以上連続する前記白画素以外の画素の画素列をランと呼び、前記主走査線内のランの始点およびその画素値、並びに、終点およびその画素値をラン情報と呼ぶとき、

入力された前記多値画像に基づいて、前記多値画像の主走査線毎に、主走査線内のラン数およびラン情報を解析するラン解析部と、

前回、前記ラン解析部により解析された直前の主走査線のラン数と、今回、前記ラン解析部により解析された主走査線のラン数とを比較する比較部と、

前記ラン解析部により解析された主走査線のラン数が1回以上連続して0である回数を表すオフセット値をカウントするカウンタと、

後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記ラン数である旨を表す符号データをラン命令と呼ぶとき、

今回、前記ラン解析部により解析された主走査線のラン数が0でない場合、

前回、前記ラン解析部により解析された直前の主走査線のラン数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、

次に、前記比較部の比較結果が不一致のとき、前記ラン命令と、前記ラン解析部により解析されたラン数と、前記ラン解析部により解析されたラン情報とを結合して符号を構成し、

前記比較部の比較結果が一致のとき、前記ラン解析部により解析されたラン情報により符号を構成する符号構成部とを備えたことを特徴とする画像符号化装置。

【請求項23】 符号化された多値画像の符号データを複数の画素からなる多値画像に復号化する画像復号化装置であって、

前記多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、

前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、

前記多値画像の主走査線内に1つ以上連続する前記白画素以外の画素の画素列をランと呼び、前記主走査線内のランの始点およびその画素値、並びに、終点およびその画素値をラン情報と呼び、

全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、

後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記ラン数である旨を表す符号データをラン命令と呼ぶとき、

入力された符号データに含まれるオフセット命令およびラン命令を解析する命令解析部と、

前記命令解析部によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、前記命令解析部によりラン命令が解析された場合、ラン命令に後続するラン数を解析し、解析されたラン数に後続するラン情報を前記ラン数毎に取り出し、取り出されたラン情報に基づいて、ランの始点および終点間の画素の画素値を補間して主走査線を復号化する復号化部とを備えたことを特徴とする画像復号化装置。

【請求項24】 請求項22記載の画像符号化装置と請求項23記載の画像復号化装置とを有することを特徴とする画像符号化復号化装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、例えば図面のように、図形を表す画素に対し、図形の背景を表す画素の割合が極めて多い画像を好適に符号化および復号化する画像符号化復号化方法及びその装置に関する。

【0002】

【従来の技術】従来、複数の白画素および黒画素からなる2値画像を、主走査線方向に連続する白ランおよび黒ランに基づいて符号化するランレングス符号化方式が知られている。ランレングス符号化方式は、ファクシミリ装置の符号化方式に適用されており、例えばG3ファクシミリ装置のMH (Modified Huffman) 符号化方式では、性質の異なる白ランおよび黒ランのランレングスに対し、それぞれ異なる符号語が割り当てられてハフマン符号が構成され、画像情報が圧縮される。ランレングス符号化方式は、主走査線内で白画素から黒画素へ、黒画素から白画素に変化する変化画素の相対的な位置を符号化していることになる。

【0003】

【発明が解決しようとする課題】しかしながら、ランレングス符号化方式を適用した従来の画像符号化復号化方法および装置にあつては、黒ランおよび白ランの双方に



符号語を割り当てて符号を構成するため、例えば図面のように、図形を表す画素に対し、図形の背景を表す画素が極めて多い画像を符号化する場合には、図形の背景を表す情報が大部分を占めることになる。図形の背景を表す情報が冗長となってしまふといった問題があった。

【0004】

【課題を解決するための手段】本発明は以上の点を解決するため次の構成を採用する。

〈構成1〉2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼び、入力された前記2値画像または前記2値画像のランレングス符号に基づいて、前記2値画像の主走査線毎に、主走査線内のラン数およびラン情報を解析するラン解析工程と、前回、前記ラン解析工程により解析された直前の主走査線のラン数と、今回、前記ラン解析工程により解析された主走査線のラン数とを比較する比較工程と、前記ラン解析工程により解析された主走査線のラン数が1回以上連続して0である回数を表すオフセット値をカウントするカウント工程と、後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが主走査線のラン数である旨を表す符号データをラン命令と呼ぶとき、今回、前記ラン解析工程により解析された主走査線のラン数が0でない場合、前回、前記ラン解析工程により解析された直前の主走査線のラン数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、次に、前記比較工程の比較結果が不一致のとき、前記ラン命令と、前記ラン解析工程により解析されたラン数と、前記ラン解析工程により解析されたラン情報とを結合して符号を構成し、前記比較工程の比較結果が一致のとき、前記ラン解析工程により解析されたラン情報により符号を構成する符号構成工程とを含むことを特徴とする画像符号化方法。

【0005】〈構成2〉符号化された2値画像の符号データを複数の画素からなる2値画像に復号化する画像復号化方法であって、前記2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼び、全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記ラン数である旨を表す符号データをラン命令と呼ぶとき、入力された符号データに含まれるオフセット命令およびラン命令を解

析する命令解析工程と、前記命令解析工程によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、前記命令解析工程によりラン命令が解析された場合、ラン命令に後続するラン数を解析し、解析されたラン数に後続するラン情報を前記ラン数毎に取り出して主走査線を復号化する復号化工程とを含むことを特徴とする画像復号化方法。

【0006】〈構成3〉構成1記載の画像符号化方法と構成2記載の画像復号化方法とを有することを特徴とする画像符号化復号化方法。

【0007】〈構成4〉2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼び、入力された前記2値画像または前記2値画像のランレングス符号に基づいて、前記2値画像の主走査線毎に、主走査線内のラン数およびラン情報を解析するラン解析部と、前回、前記ラン解析部により解析された直前の主走査線のラン数と、今回、前記ラン解析部により解析された主走査線のラン数とを比較する比較部と、前記ラン解析部により解析された主走査線のラン数が1回以上連続して0である回数を表すオフセット値をカウントするカウンタと、後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが主走査線のラン数である旨を表す符号データをラン命令と呼ぶとき、今回、前記ラン解析部により解析された主走査線のラン数が0でない場合、前回、前記ラン解析部により解析された直前の主走査線のラン数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、次に、前記比較部の比較結果が不一致のとき、前記ラン命令と、前記ラン解析部により解析されたラン数と、前記ラン解析部により解析されたラン情報とを結合して符号を構成し、前記比較部の比較結果が一致のとき、前記ラン解析部により解析されたラン情報により符号を構成する符号構成部とを備えたことを特徴とする画像符号化装置。

【0008】〈構成5〉符号化された2値画像の符号データを複数の画素からなる2値画像に復号化する画像復号化装置であって、前記2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼び、全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記ラン数である旨



を表す符号データをラン命令と呼ぶとき、入力された符号データに含まれるオフセット命令およびラン命令を解析する命令解析部と、前記命令解析部によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、前記命令解析部によりラン命令が解析された場合、ラン命令に後続するラン数を解析し、解析されたラン数に後続するラン情報を前記ラン数毎に取り出して主走査線を復号化する復号化部とを備えたことを特徴とする画像復号化装置。

【0009】〈構成6〉構成4記載の画像符号化装置と構成5記載の画像復号化装置とを有することを特徴とする画像符号化復号化装置。

【0010】〈構成7〉2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼ぶとき、入力された前記2値画像または前記2値画像のランレングス符号に基づいて、前記2値画像の主走査線毎に、主走査線内のラン数およびラン情報を解析するラン解析工程と、前回、前記ラン解析工程により解析された直前の主走査線のラン数と、今回、前記ラン解析工程により解析された主走査線のラン数との差を算出する算出工程と、前記ラン解析工程により解析された主走査線のラン数が1回以上連続して0である回数を表すオフセット値をカウントするカウント工程と、後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、符号化すべき主走査線のラン数が直前の主走査線のラン数より多い旨を表す符号データを進入命令と呼び、符号化すべき主走査線のラン数が直前の主走査線のラン数より少ない旨を表す符号データを退出命令と呼ぶとき、今回、前記ラン解析工程により解析された主走査線のラン数が0でない場合、前回、前記ラン解析工程により解析された直前の主走査線のラン数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、次に、前記算出工程の算出結果が符号化すべき主走査線のラン数が直前の主走査線のラン数より多い旨を意味するとき、前記進入命令と、前記算出工程の算出結果の絶対値と、前記ラン解析工程により解析されたラン情報とを結合して符号を構成し、前記算出工程の算出結果が符号化すべき主走査線のラン数が直前の主走査線のラン数より少ない旨を意味するとき、前記退出命令と、前記算出工程の算出結果の絶対値と、前記ラン解析工程により解析されたラン情報とを結合して符号を構成し、前記算出工程の算出結果が符号化すべき主走査線のラン数と直前の主走査線のラン数とが同数である旨を意味するとき、前記ラン解析工程により解析されたラン情報により符号を構成する符号構成工程とを含むことを特徴とする画像符号化方法。

【0011】〈構成8〉符号化された2値画像の符号デ

ータを複数の画素からなる2値画像に復号化する画像復号化方法であって、前記2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼び、全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、後続する符号データがオフセット値である旨を表す符号データをオフセット命令と呼び、復号化すべき主走査線のラン数が直前の主走査線より多い旨を表す符号データを進入命令と呼び、復号化すべき主走査線のラン数が直前の主走査線より少ない旨を表す符号データを退出命令と呼び、復号化すべき主走査線のラン数と直前の主走査線のラン数との差の絶対値を差分絶対値と呼ぶとき、入力された符号データに含まれるオフセット命令、進入命令および退出命令を解析する命令解析工程と、既に復号化済みの直前の主走査線のラン数を保持し、前記命令解析工程により進入命令が解析された場合、進入命令に後続する前記差分絶対値を、直前の主走査線のラン数に加算し、前記命令解析工程により退出命令が解析された場合、退出命令に後続する前記差分絶対値を、直前の主走査線のラン数から減算し、復号化すべき主走査線に含まれるラン数を算出する算出工程と、前記命令解析工程によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、前記命令解析工程により進入命令が解析された場合および前記命令解析工程により退出命令が解析された場合、前記差分絶対値に後続するラン情報を、前記算出工程により算出されたラン数毎に取り出して主走査線を復号化する復号化工程とを含むことを特徴とする画像復号化方法。

【0012】〈構成9〉構成7記載の画像符号化方法と構成8記載の画像復号化方法とを有することを特徴とする画像符号化復号化方法。

【0013】〈構成10〉2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼ぶとき、入力された前記2値画像または前記2値画像のランレングス符号に基づいて、前記2値画像の主走査線毎に、主走査線内のラン数およびラン情報を解析するラン解析部と、前回、前記ラン解析部により解析された直前の主走査線のラン数と、今回、前記ラン解析部により解析された主走査線のラン数との差を算出する算出部と、前記ラン解析部により解析された主走査線のラン数が1回以上連続して0である回数を表すオフセット値をカウントするカウンタと、後続する符号データが前記オフセット値である旨を表す符号データをオフセット

命令と呼び、符号化すべき主走査線のラン数が直前の主走査線のラン数より多い旨を表す符号データを進入命令と呼び、符号化すべき主走査線のラン数が直前の主走査線のラン数より少ない旨を表す符号データを退出命令と呼ぶとき、今回、前記ラン解析部により解析された主走査線のラン数が0の場合、前回、前記ラン解析部により解析された直前の主走査線のラン数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、次に、前記算出部の算出結果が符号化すべき主走査線のラン数が直前の主走査線のラン数より多い旨を意味するとき、前記進入命令と、前記算出部の算出結果の絶対値と、前記ラン解析部により解析されたラン情報とを結合して符号を構成し、前記算出部の算出結果が符号化すべき主走査線のラン数が直前の主走査線のラン数より少ない旨を意味するとき、前記退出命令と、前記算出部の算出結果の絶対値と、前記ラン解析部により解析されたラン情報とを結合して符号を構成し、前記算出部の算出結果が符号化すべき主走査線のラン数と直前の主走査線のラン数とが同数である旨を意味するとき、前記ラン解析部により解析されたラン情報により符号を構成する符号構成部とを備えたことを特徴とする画像符号化装置。

【0014】〈構成11〉符号化された2値画像の符号データを複数の画素からなる2値画像に復号化する画像復号化装置であって、前記2値画像が、図形を表す複数の黒画素と、図形の背景を表す複数の白画素とを含み、前記2値画像の主走査線内に1つ以上連続する黒画素の画素列をランと呼び、前記主走査線内のランの始点および終点、または、始点および画素数を表す情報をラン情報と呼び、全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、後続する符号データがオフセット値である旨を表す情報をオフセット命令と呼び、復号化すべき主走査線のラン数が直前の主走査線より多い旨を表す情報を進入命令と呼び、復号化すべき主走査線のラン数が直前の主走査線より少ない旨を表す情報を退出命令と呼び、復号化すべき主走査線のラン数と直前の主走査線のラン数との差の絶対値を差分絶対値と呼ぶとき、入力された符号データに含まれるオフセット命令、進入命令および退出命令を解析する命令解析部と、既に復号化済みの直前の主走査線のラン数を保持し、前記命令解析部により進入命令が解析された場合、進入命令に後続する前記差分絶対値を、直前の主走査線のラン数に加算し、前記命令解析部により退出命令が解析された場合、退出命令に後続する前記差分絶対値を、直前の主走査線のラン数から減算して復号化すべき主走査線に含まれるラン数を算出する算出部と、前記命令解析部によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、前記命令解析部により進入命令が解析された場

合および前記命令解析部により退出命令が解析された場合、前記差分絶対値に後続するラン情報を、前記算出部により算出されたラン数毎に取り出して主走査線を復号化する復号化部とを備えたことを特徴とする画像復号化装置。

【0015】〈構成12〉構成10記載の画像符号化装置と構成11記載の画像復号化装置とを有することを特徴とする画像符号化復号化装置。

【0016】〈構成13〉多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、前記多値画像の主走査線内に1つ以上連続する第1画素の画素列を第1ランと呼び、前記多値画像の主走査線内に1つ以上連続する第2画素の画素列を第2ランと呼び、前記主走査線内の第1ランの始点、終点および第1画素値、または、始点、画素数および第1画素値を表す情報を第1ラン情報と呼び、前記主走査線内の第2ランの始点、終点および第2画素値、または、始点、画素数および第2画素値を表す情報を第2ラン情報と呼び、入力された前記多値画像に基づいて、前記多値画像の主走査線毎に、主走査線内の第1および第2ランの総数、並びに、第1および第2ラン情報を解析するラン解析工程と、前回、前記ラン解析工程により解析された直前の主走査線の第1および第2ランの総数と、今回、前記ラン解析工程により解析された主走査線の第1および第2ランの総数とを比較する比較工程と、前記ラン解析工程により解析された主走査線の第1および第2ランの総数が1回以上連続して0である回数を表すオフセット値をカウントするカウント工程と、後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記第1および第2ランの総数である旨を表す符号データをラン命令と呼ぶとき、今回、前記ラン解析工程により解析された主走査線の第1および第2ランの総数が0でない場合、前回、前記ラン解析工程により解析された直前の主走査線の第1および第2ランの総数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、次に、前記比較工程の比較結果が不一致のとき、前記ラン命令と、前記ラン解析工程により解析された第1および第2ランの総数と、前記ラン解析工程により解析された第1および第2ラン情報とを結合して符号を構成し、前記比較工程の比較結果が一致のとき、前記ラン解析工程により解析された第1および第2ラン情報により符号を構成する符号構成工程とを含むことを特徴とする画像符号化方法。

【0017】〈構成14〉符号化された多値画像の符号データを複数の画素からなる多値画像に復号化する画像復号化方法であって、前記多値画像が、図形を表す第1

画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、前記多値画像の主走査線内に1つ以上連続する第1画素の画素列を第1ランと呼び、前記多値画像の主走査線内に1つ以上連続する第2画素の画素列を第2ランと呼び、前記主走査線内の第1ランの始点、終点および第1画素値、または、始点、画素数および第1画素値を表す情報を第1ラン情報と呼び、前記主走査線内の第2ランの始点、終点および第2画素値、または、始点、画素数および第2画素値を表す情報を第2ラン情報と呼び、全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記第1および第2ランの総数である旨を表す符号データをラン命令と呼ぶとき、入力された符号データに含まれるオフセット命令およびラン命令を解析する命令解析工程と、前記命令解析工程によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、前記命令解析工程によりラン命令が解析された場合、ラン命令に後続する第1および第2ランの総数を解析し、解析された第1および第2ランの総数に後続する第1および第2ラン情報を前記第1および第2ランの総数毎に取り出して主走査線を復号化する復号化工程とを含むことを特徴とする画像復号化方法。

【0018】〈構成15〉構成13記載の画像符号化方法と構成14記載の画像復号化方法とを有することを特徴とする画像符号化復号化方法。

【0019】〈構成16〉多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、前記多値画像の主走査線内に1つ以上連続する第1画素の画素列を第1ランと呼び、前記多値画像の主走査線内に1つ以上連続する第2画素の画素列を第2ランと呼び、前記主走査線内の第1ランの始点、終点および第1画素値、または、始点、画素数および第1画素値を表す情報を第1ラン情報と呼び、前記主走査線内の第2ランの始点、終点および第2画素値、または、始点、画素数および第2画素値を表す情報を第2ラン情報と呼び、入力された前記多値画像に基づいて、前記多値画像の主走査線毎に、主走査線内の第1および第2ランの総数、並びに、第1および第2ラン情報を解析するラン解析部と、前回、前記ラン解析部により解析された直前の主走査線の第1および第2ランの総数と、今回、前記ラン解析部により解析された主走査線の第1および第2ランの総数

とを比較する比較部と、前記ラン解析部により解析された主走査線の第1および第2ランの総数が1回以上連続して0である回数を表すオフセット値をカウントするカウンタと、後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記第1および第2ランの総数である旨を表す符号データをラン命令と呼ぶとき、今回、前記ラン解析部により解析された主走査線の第1および第2ランの総数が0でない場合、前回、前記ラン解析部により解析された直前の主走査線の第1および第2ランの総数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、次に、前記比較部の比較結果が不一致のとき、前記ラン命令と、前記ラン解析部により解析された第1および第2ランの総数と、前記ラン解析部により解析された第1および第2ラン情報を結合して符号を構成し、前記比較部の比較結果が一致のとき、前記ラン解析部により解析された第1および第2ラン情報により符号を構成する符号構成部とを備えたことを特徴とする画像符号化装置。

【0020】〈構成17〉符号化された多値画像の符号データを複数の画素からなる多値画像に復号化する画像復号化装置であって、前記多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、前記多値画像の主走査線内に1つ以上連続する第1画素の画素列を第1ランと呼び、前記多値画像の主走査線内に1つ以上連続する第2画素の画素列を第2ランと呼び、前記主走査線内の第1ランの始点、終点および第1画素値、または、始点、画素数および第1画素値を表す情報を第1ラン情報と呼び、前記主走査線内の第2ランの始点、終点および第2画素値、または、始点、画素数および第2画素値を表す情報を第2ラン情報と呼び、全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記第1および第2ランの総数である旨を表す符号データをラン命令と呼ぶとき、入力された符号データに含まれるオフセット命令およびラン命令を解析する命令解析部と、前記命令解析部によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、前記命令解析部によりラン命令が解析された場合、ラン命令に後続する第1および第2ランの総数を解析し、解析された第1および第2ランの総数に後続する第1および第2ラン情報を前記第1および第2ランの総数毎に取り出して主走査線を復号化する復号化部とを備えたことを特徴とする画像復号化装置。

【0021】〈構成18〉構成16記載の画像符号化装置と構成17記載の画像復号化装置とを有することを特徴とする画像符号化復号化装置。

【0022】〈構成19〉多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、前記多値画像の主走査線内に1つ以上連続する前記白画素以外の画素の画素列をランと呼び、前記主走査線内のランの始点およびその画素値、並びに、終点およびその画素値をラン情報と呼ぶとき、入力された前記多値画像に基づいて、前記多値画像の主走査線毎に、主走査線内のラン数およびラン情報を解析するラン解析工程と、前回、前記ラン解析工程により解析された直前の主走査線のラン数と、今回、前記ラン解析工程により解析された主走査線のラン数とを比較する比較工程と、前記ラン解析工程により解析された主走査線のラン数が1回以上連続して0である回数を表すオフセット値をカウントするカウント工程と、後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記ラン数である旨を表す符号データをラン命令と呼ぶとき、今回、前記ラン解析工程により解析された主走査線のラン数が0でない場合、前回、前記ラン解析工程により解析された直前の主走査線のラン数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、次に、前記比較工程の比較結果が不一致のとき、前記ラン命令と、前記ラン解析工程により解析されたラン数と、前記ラン解析工程により解析されたラン情報とを結合して符号を構成し、前記比較工程の比較結果が一致のとき、前記ラン解析工程により解析されたラン情報により符号を構成する符号構成工程とを含むことを特徴とする画像符号化方法。

【0023】〈構成20〉符号化された多値画像の符号データを複数の画素からなる多値画像に復号化する画像復号化方法であって、前記多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、前記多値画像の主走査線内に1つ以上連続する前記白画素以外の画素の画素列をランと呼び、前記主走査線内のランの始点およびその画素値、並びに、終点およびその画素値をラン情報と呼び、全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記ラン数である旨を表す符号データをラン命令と呼

ぶとき、入力された符号データに含まれるオフセット命令およびラン命令を解析する命令解析工程と、前記命令解析工程によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、前記命令解析工程によりラン命令が解析された場合、ラン命令に後続するラン数を解析し、解析されたラン数に後続するラン情報を前記ラン数毎に取り出し、取り出されたラン情報に基づいて、ランの始点および終点間の画素の画素値を補間して主走査線を復号化する復号化工程とを含むことを特徴とする画像復号化方法。

【0024】〈構成21〉構成19記載の画像符号化方法と構成20記載の画像復号化方法とを有することを特徴とする画像符号化復号化方法。

【0025】〈構成22〉多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそれぞれ異なる第3画素値を有する複数の白画素とを含み、前記多値画像の主走査線内に1つ以上連続する前記白画素以外の画素の画素列をランと呼び、前記主走査線内のランの始点およびその画素値、並びに、終点およびその画素値をラン情報と呼ぶとき、入力された前記多値画像に基づいて、前記多値画像の主走査線毎に、主走査線内のラン数およびラン情報を解析するラン解析部と、前回、前記ラン解析部により解析された直前の主走査線のラン数と、今回、前記ラン解析部により解析された主走査線のラン数とを比較する比較部と、前記ラン解析部により解析された主走査線のラン数が1回以上連続して0である回数を表すオフセット値をカウントするカウンタと、後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記ラン数である旨を表す符号データをラン命令と呼ぶとき、今回、前記ラン解析部により解析された主走査線のラン数が0でない場合、前回、前記ラン解析部により解析された直前の主走査線のラン数が0のとき、前記オフセット命令と前記オフセット値とを結合して符号を構成し、次に、前記比較部の比較結果が不一致のとき、前記ラン命令と、前記ラン解析部により解析されたラン数と、前記ラン解析部により解析されたラン情報とを結合して符号を構成し、前記比較部の比較結果が一致のとき、前記ラン解析部により解析されたラン情報により符号を構成する符号構成部とを備えたことを特徴とする画像符号化装置。

【0026】〈構成23〉符号化された多値画像の符号データを複数の画素からなる多値画像に復号化する画像復号化装置であって、前記多値画像が、図形を表す第1画素値を有する複数の第1画素と、図形を表し、前記第1画素値と異なる第2画素値を有する複数の第2画素と、図形の背景を表し、前記第1および第2画素値とそ

れぞれ異なる第3画素値を有する複数の白画素とを含み、前記多値画像の主走査線内に1つ以上連続する前記白画素以外の画素の画素列をランと呼び、前記主走査線内のランの始点およびその画素値、並びに、終点およびその画素値をラン情報と呼び、全画素が前記白画素により構成された主走査線を白ラインと呼び、副走査方向に1つ以上連続する白ライン数をオフセット値と呼び、後続する符号データが前記オフセット値である旨を表す符号データをオフセット命令と呼び、後続する符号データが前記ラン数である旨を表す符号データをラン命令と呼ぶとき、入力された符号データに含まれるオフセット命令およびラン命令を解析する命令解析部と、前記命令解析部によりオフセット命令が解析された場合、オフセット命令に後続するオフセット値に相当する白ラインを復号化し、前記命令解析部によりラン命令が解析された場合、ラン命令に後続するラン数を解析し、解析されたラン数に後続するラン情報を前記ラン数毎に取り出し、取り出されたラン情報に基づいて、ランの始点および終点間の画素の画素値を補間して主走査線を復号化する復号化部とを備えたことを特徴とする画像復号化装置。

【0027】〈構成24〉構成22記載の画像符号化装置と構成23記載の画像復号化装置とを有することを特徴とする画像符号化復号化装置。

【0028】

【発明の実施の形態】以下、本発明の実施の形態を具体例を用いて説明する。

《具体例1》

〈画像符号化方法および復号化方法〉図1は本発明に係る具体例1の画像符号化方法の概略説明図であり、図1(a)は符号化対象の2値画像の一例を示す図、図1(b)は図1(a)に示された2値画像の符号構成を示す図である。図1(a)に示すように、2値画像の主走査線方向の座標をxで表し、副走査方向の座標をyで表す。符号化対象の2値画像のサイズは任意であるが、説明を容易にするため、8行8列の画素からなるものとする。図中、斜線で示される画素は図形を構成する黒画素であり、白地で示される画素は図形の背景を構成する白画素である。なお、白画素が図形を構成し、黒画素が図形の背景を構成するようにしてもよい。

【0029】ところで、一般に主走査線方向に1つ以上連続する同一画素値の画素列をランといい、この画素列を構成する画素数をラン長またはランレングスという。以下、主走査方向に1つ以上連続する黒画素の画素列を黒ランといい、1本の主走査線内の黒ランの数を黒ラン数という。また、全画素が白画素により構成された主走査線を白ラインという。図において、R1～R6が黒ランである。

【0030】具体例1の画像符号化方法は、主走査線内の黒ラン数をカウントし、カウントされた黒ラン数に基づいて2値画像を符号化するものである。主走査線内に

黒ラン数がカウントされない場合、すなわち主走査線が白ラインの場合には、副走査方向に1つ以上連続する白ラインの数を表すオフセット値をカウントし、カウントされたオフセット値そのものを符号とする。主走査線内に黒ラン数がカウントされた場合には、直前の主走査線の黒ラン数と比較し、直前の主走査線の黒ラン数と一致しないときには、カウントされた黒ラン数および対応する黒ランの位置情報を符号化し、直前の主走査線の黒ラン数と一致したときには、対応する黒ランの位置情報を符号化する。

【0031】黒ランの位置情報は、主走査線内の黒ランの始点および終点の画素位置、または、主走査線内の黒ランの始点およびラン長により構成される。オフセット値の前には、後続する符号がオフセット値である旨を表すオフセット命令が挿入され、黒ラン数の前には、後続する符号が黒ラン数である旨を表すラン命令が挿入される。すなわち、この画像符号化方法は、同数の黒ラン数が連続する主走査線のそれぞれの黒ランの位置情報を順次に結合して、図形のグループ化を表すセグメントを構成し、2値画像を符号化するものである。画像復号化方法は、符号に含まれるオフセット命令およびラン命令を解析し、オフセット命令が解析された場合には、後続するオフセット値に相当する白ラインを復号化し、ラン命令が解析された場合には、後続する黒ラン数を解析し、解析された黒ラン数毎に、黒ラン数に後続する黒ランの位置情報を取り出し、主走査線を復号化する。

【0032】次に図1(b)に示された符号構成について具体的に説明する。図1(b)において、まず、y座標0～2の主走査線は、白ラインなので、オフセット命令にオフセット値「3」が結合される。次いで、y座標3の主走査線では、黒ラン数が1なので、ラン命令が挿入され、黒ラン数「1」、並びに、黒ランR1の始点「2」および終点「3」が順次に結合される。次いで、y座標4の主走査線では、黒ラン数が直前のラインの黒ラン数と同数のなので、黒ランR2の始点「2」および終点「3」が結合される。次いで、y座標5の主走査線では、黒ラン数が2に変化したので、ラン命令が挿入され、黒ラン数「2」が結合され、黒ランR3の始点「2」および終点「3」、並びに、黒ランR4の始点「6」および終点「7」が順次に結合される。次いで、y座標6の主走査線では、黒ラン数が直前のラインの黒ラン数と同数のなので、黒ランR5の始点「2」および終点「3」、並びに、黒ランR6の始点「6」および終点「7」が順次に結合される。次いで、y座標7および8の主走査線は、白ラインなので、オフセット命令が挿入され、オフセット値「2」が結合される。

【0033】このように、具体例1の画像符号化方法によれば、副走査方向に1つ以上連続し、黒ラン数が同数の主走査線をグループ化し、主走査線内の黒ラン数が0の場合には、オフセット値をカウントし、オフセット命

令にオフセット値を結合して符号を構成するとともに、主走査線内の黒ラン数が0でない場合には、ラン命令に黒ラン数および黒ランの位置情報を結合して符号を構成する。したがって、白画素の情報量を極めて少なくすることができるので、高い圧縮率の符号を得ることができる。また、複雑な演算を行うことなく2値画像を符号化することができるので、符号化処理を高速に行うことができる。

【0034】また、具体例1の画像復号化方法によれば、オフセット命令およびラン命令を解析し、符号化された2値画像を好適に復号化することができる。また、複雑な演算を行うことなく符号化された2値画像を復号化することができるので、復号化処理を高速に行うことができる。具体例1の画像符号化方法および画像復号化方法は、特に、図面のように、図形を表す画素に対し、図形の背景を表す画素の割合が極めて多い画像に対して有効である。

【0035】〈画像符号化装置〉図2は本発明に係る具体例1の画像符号化装置の構成を示すブロック図である。図2に示すように、画像符号化装置は、画像メモリ11、黒ラン解析部12、黒ラン位置バッファ15、黒ラン数比較部16、オフセット値カウンタ17、符号構成部18および符号メモリ19を備えている。

【0036】画像メモリ11には、符号化対象の2値画像が記憶されている。黒ラン解析部12は、画像メモリ11に記憶された2値画像を走査順に読み出し、読み出された走査線内の黒ランを解析するものであり、黒ラン位置解析部13および黒ラン数カウンタ14を有する。黒ラン位置解析部13は、走査線内の黒ランの始点および終点を解析し、解析された始点および終点を黒ランの位置情報として順次黒ラン位置バッファ15に出力する。黒ラン数カウンタ14は、走査線内の黒ラン数をカウントし、カウントされた黒ラン数を黒ラン数比較部16、オフセット値カウンタ17および符号構成部18に出力する。

【0037】黒ラン位置バッファ15は、黒ラン位置解析部13から出力された主走査線内の黒ランの位置情報を一時的に保持するバッファである。黒ラン数比較部16は、前回、黒ラン数カウンタ14によりカウントされた直前の主走査線の黒ラン数を保持する図示しない保持部を有し、前回、黒ラン数カウンタ14によりカウントされた主走査線の黒ラン数と、今回、黒ラン数カウンタ14によりカウントされた主走査線の黒ラン数とを比較する。黒ラン数比較部16は、入力された黒ラン数と直前の主走査線の黒ラン数とが一致した場合には、一致信号を符号構成部18に出力し、入力された黒ラン数と直前の主走査線の黒ラン数とが一致しない場合には、不一致信号を符号構成部18に出力する。

【0038】オフセット値カウンタ17は、黒ラン数カウンタ14にカウントされた黒ラン数に基づいて、副走

査方向に1つ以上連続する白ラインの数を表すオフセット値をカウントする。オフセット値カウンタ17は、入力された黒ラン数が0の場合には、オフセット値を1インクリメントし、入力された黒ラン数が0でない場合には、オフセット値を参照し、参照されたオフセット値が0でないとき、オフセット値を符号構成部18に出力する。

【0039】符号構成部18は、前回、黒ラン数カウンタ14によりカウントされた黒ラン数を保持する図示しない保持部を有し、今回、黒ラン数カウンタ14によりカウントされた黒ラン数に基づいて符号を構成する。符号構成部18は、今回、入力された黒ラン数が0でない場合には、前回、入力された黒ラン数を参照し、参照された前回の黒ラン数が0のとき、オフセット値カウンタ17により出力されたオフセット値の前にオフセット命令を挿入して符号を構成し、参照された前回の黒ラン数が0でないとき、黒ラン数比較部16により出力された信号を参照する。

【0040】符号構成部18は、黒ラン数比較部16から出力された信号が不一致信号の場合には、今回、入力された黒ラン数の前にラン命令を挿入するとともに、黒ラン位置バッファ15に保持された黒ランの位置情報を読み出し、今回、入力された黒ラン数に結合して符号を構成する。符号構成部18は、黒ラン数比較部16から出力された信号が一致信号の場合には、黒ラン位置バッファ15に保持されたランの位置情報を読み出し、そのまま符号を構成する。符号構成部18により構成された符号は、符号メモリ19に記憶される。

【0041】図3および図4は図2に示された具体例1の画像符号化装置の動作を示すフローチャートである。まず、画像メモリ11に記憶された2値画像が走査順に1走査線毎に黒ラン解析部12に読み出され（ステップS1）、黒ラン位置解析部13により主走査線内の黒ランの位置情報が解析されて黒ラン位置バッファ15に保持されるとともに、黒ラン数カウンタ14により主走査線内の黒ラン数がカウントされる（ステップS2）。

【0042】次いで、黒ラン数比較部16により、前回、黒ラン数カウンタ14によりカウントされた直前の主走査線の黒ラン数と、今回、黒ラン数カウンタ14によりカウントされた黒ラン数とが比較され（ステップS3）、比較結果が一致の場合には、一致信号が符号構成部18に出力され（ステップS4）、比較結果が不一致の場合には、不一致信号が符号構成部18に出力される（ステップS5）。次いで、オフセット値カウンタ17では、今回、黒ラン数カウンタ14によりカウントされた黒ラン数が0か否かが判定される（ステップS6）。ステップS6で、黒ラン数が0の場合には、オフセット値が1インクリメントされ（ステップS7）、ステップS16に進む。一方、ステップS6で、黒ラン数が0でない場合には、オフセット値が参照される（ステップS



8)。ここで、オフセット値が0でない場合には、オフセット値が符号構成部18に出力され、オフセット値が0にリセットされ(ステップS9)、ステップS10に進む。一方、オフセット値が0の場合には、ステップS10に進む。

【0043】符号構成部18では、今回、黒ラン数カウンタ14によりカウントされた黒ラン数が0か否かが判定される(ステップS10)。ステップS10で、今回、黒ラン数カウンタ14によりカウントされた黒ラン数が0の場合には、ステップS16に進む。一方、ステップS10で、今回、黒ラン数カウンタ14によりカウントされた黒ラン数が0でない場合には、前回、黒ラン数カウンタ14によりカウントされた直前の主走査線の黒ラン数が0か否かが判定される(ステップS11)。ステップS11で、直前の主走査線の黒ラン数が0でない場合には、ステップS13に進む。一方、ステップS13で、直前の主走査線の黒ラン数が0のときには、オフセット値カウンタ17により出力されたオフセット値の前にオフセット命令が挿入され、符号が構成される(ステップS12)。

【0044】次いで、黒ラン数比較部16から出力された信号が一致信号か不一致信号かが判定される(ステップS13)。ステップS13で、入力された信号が不一致信号の場合には、今回、黒ラン数カウンタ14によりカウントされた黒ラン数の前にラン命令が挿入され、黒ラン位置バッファ15に保持された黒ランの位置情報が読み出され、黒ラン数に結合されて符号が構成される(ステップS14)。一方、ステップS12で、入力された信号が一致信号の場合には、黒ラン位置バッファ15に保持された黒ランの位置情報が読み出され、そのまま符号が構成される(ステップS15)。

【0045】次いで、ステップS16において、次の主走査線の有無が判定され、次の主走査線がある場合には、ステップS1に戻る。一方、ステップS16で、次の走査線が無い場合には、オフセット値カウンタ17のオフセット値が0か否かが判定され(ステップS17)、オフセット値が0でない場合には、符号構成部18により、このオフセット値の前にオフセット命令が挿入されて符号が構成され、処理を終了する。一方、ステップS17で、オフセット値が0の場合には、処理を終了する。

【0046】このように、具体例1の画像符号化装置によれば、黒ラン数カウンタ14によりカウントされた黒ラン数が0の場合には、オフセット命令をカウントし、黒ラン数カウンタ14によりカウントされた黒ラン数が0でない場合には、オフセット値が0でないとき、オフセット命令にオフセット値を結合して符号を構成し、次に、黒ラン数比較部16の比較結果が不一致のとき、ラン命令に黒ラン数および黒ランの位置情報を結合して符号を構成し、黒ラン数比較部16の比較結果が一致のと

き、黒ランの位置情報をそのまま符号化する。したがって、上記の具体例1の画像符号化方法を好適に実現することができる。

【0047】〈画像復号化装置〉図5は本発明に係る具体例1の画像復号化装置の構成を示すブロック図である。図5に示すように、画像復号化装置は、図2に示された画像符号化装置により符号化された2値画像を復号化するものであり、符号メモリ21、復号化部22および画像メモリ24を備えている。

【0048】符号メモリ21には、復号化対象の符号データが記憶されている。復号化部22は、符号メモリ21に記憶された符号データをその先頭から読み出し、復号化するものであり、命令解析部23および復号化すべき主走査線の黒ラン数を保持する図示しない保持部を有する。命令解析部23は、入力された符号データに含まれるオフセット命令およびラン命令を解析する。復号化部22は、命令解析部23によりオフセット命令が解析された場合には、後続するオフセット値を解析し、解析されたオフセット値に相当する白ラインを復号化する。復号化部22は、命令解析部23によりラン命令が解析された場合には、後続する黒ラン数を解析して図示しない保持部に保持し、保持された黒ラン数毎に、黒ラン数に後続する黒ランの位置情報(黒ランの始点および終点)を取り出し、主走査線を復号化する。画像メモリ24には、復号化部22により復号化された2値画像が記憶される。

【0049】図6は図5に示された具体例1の画像復号化装置の動作を示すフローチャートである。まず、符号メモリ21に記憶された符号データがその先頭から読み出され、復号化部22に入力される(ステップS21)。次いで、復号化部22の命令解析部23により入力された符号に含まれるオフセット命令が解析される(ステップS22)。ステップS22で、オフセット命令が解析された場合には、オフセット命令に後続するオフセット値が解析され、解析されたオフセット値に相当する白ラインが復号化され(ステップS23)、ステップS27に進む。

【0050】一方、ステップS22において、オフセット命令が解析されない場合には、符号データに含まれるラン命令が解析される(ステップS24)。ステップS24において、ラン命令が解析された場合には、ラン命令に後続する黒ラン数が解析され図示しない保持部に保持され、保持された黒ラン数に後続する黒ランの位置情報(黒ランの始点および終点)が取り出され、主走査線が復号化される(ステップS25)。一方、ステップS24において、ラン命令が解析されない場合には、ステップS26に進み、保持部に保持された黒ラン数の後続する黒ランの位置情報が取り出され、主走査線が復号化される。次いで、ステップS27において、次の符号データがある場合には、ステップS21に戻り、次の符号



データがない場合には、処理を終了する。

【0051】このように、具体例1の画像復号化装置によれば、命令解析部23のよりオフセット命令が解析された場合には、後続するオフセット値に相当する白ラインを復号化し、ラン命令が解析された場合には、ラン数毎に黒ランの位置情報を取り出し、主走査線を復号化する。したがって、上記の具体例1の画像復号化方法を好適に実現することができる。

【0052】《具体例2》

〈画像符号化方法および復号化方法〉図7は本発明に係る具体例2の画像符号化方法の概略説明図であり、図7(a)は符号化対象の2値画像の一例を示す図、図7(b)は図7(a)に示された2値画像の符号構成を示す図である。図7(a)に示すように、2値画像の主走査線方向の座標をxで表し、副走査方向の座標をyで表す。符号化対象の2値画像のサイズは任意であるが、説明を容易にするため、8行8列の画素からなるものとする。図中、斜線で示される画素は図形を構成する黒画素であり、白地で示される画素は図形の背景を構成する白画素である。なお、白画素が図形を構成し、黒画素が図形の背景を構成するようにしてもよい。また、具体例1と同様に、主走査方向に1つ以上連続する黒画素の画素列を黒ランといい、1本の主走査線内の黒ランの数を黒ラン数という。また、全画素が白画素により構成された主走査線を白ラインという。図において、R11~R18が黒ランである。

【0053】具体例2の画像符号化方法は、具体例1の画像符号化方法と同様に、主走査線内の黒ラン数をカウントし、カウントされた黒ラン数に基づいて2値画像を符号化するものである。主走査線内に黒ラン数がカウントされない場合、すなわち主走査線が白ラインの場合には、副走査方向に1つ以上連続する白ラインの数を表すオフセット値をカウントし、カウントされたオフセット値そのものを符号とする。主走査線内に黒ラン数がカウントされた場合には、今回、カウントされた主走査線の黒ラン数と、前回、カウントされた直前の主走査線の黒ラン数との差を算出する。そして、今回カウントされた主走査線の黒ラン数が前回カウントされた直前の主走査線に黒ラン数より多い場合には、黒ラン数の増分値(差の絶対値)および対応する黒ランの位置情報を符号化し、今回カウントされた主走査線の黒ラン数が前回カウントされた直前の主走査線に黒ラン数より少ない場合には、黒ラン数の減分値(差の絶対値)および対応する黒ランの位置情報を符号化し、今回カウントされた主走査線の黒ラン数と前回カウントされた直前の主走査線に黒ラン数より黒ラン数の差が0の場合には、対応する黒ランの位置情報を符号化する。

【0054】黒ランの位置情報は、主走査線内の黒ランの始点および終点の画素位置、または、主走査線内の黒ランの始点およびラン長により構成される。オフセット

値の前には、後続する符号がオフセット値である旨を表すオフセット命令が挿入される。黒ラン数の増分値の前には、後続する符号が黒ラン数の増分値である旨を表す進入命令が挿入され、黒ラン数の減分値の前には、後続する符号が黒ラン数の減分値である旨を表す退出命令が挿入される。

【0055】具体例2の画像符号化方法は、主走査線の黒ラン数を表す際に、直前の主走査線の黒ラン数との差を符号化する。黒ラン数の差を符号化することで、黒ラン数を直接符号化するよりも符号長を短くすることができる。一般に、画像内の画素は、その近傍の画素と相関が高く、ある主走査線の黒ラン数とその直前の主走査線の黒ラン数は同数に近くなるからである。

【0056】画像復号化方法は、符号に含まれるオフセット命令、進入命令および退出命令を解析し、復号化すべき黒ラン数を算出する。オフセット命令が解析された場合には、後続するオフセット値に相当する白ラインを復号化する。進入命令が解析された場合には、後続する黒ラン数の増分値を解析し、直前の主走査線の黒ラン数に、解析された黒ラン数の増分値を加算して、復号化すべき主走査線の黒ラン数を算出する。そして、算出された黒ラン数毎に、黒ラン数の増分値に後続する黒ランの位置情報を取り出し、主走査線を復号化する。退出命令が解析された場合には、後続する黒ラン数の減分値を解析し、直前の主走査線の黒ラン数から、解析された黒ラン数の減分値を減算して、復号化すべき主走査線の黒ラン数を算出する。そして、算出された黒ラン数毎に、黒ラン数の減分値に後続する黒ランの位置情報を取り出し、主走査線を復号化する。

【0057】次に図7(b)に示された符号構成について具体的に説明する。図7(b)において、まず、y座標0および1の主走査線は、白ラインなので、オフセット命令にオフセット値「2」が結合される。次いで、y座標2の主走査線では、黒ラン数が1なので、進入命令が挿入され、増分値「1」、並びに、黒ランR11の始点「2」および終点「3」が順次に結合される。次いで、y座標3の主走査線では、黒ラン数が直前のラインの黒ラン数と同数のなので、黒ランR12の始点「2」および終点「3」が結合される。次いで、y座標4の主走査線では、黒ラン数が2に増えたので、進入命令が挿入され、増分値「1」が結合され、黒ランR13の始点「2」および終点「3」、並びに、黒ランR14の始点「6」および終点「7」が順次に結合される。次いで、y座標5の主走査線では、黒ラン数が直前のラインの黒ラン数と同数のなので、黒ランR15の始点「2」および終点「3」、並びに、黒ランR16の始点「6」および終点「7」が順次に結合される。次いで、y座標6の主走査線では、黒ラン数が1に減ったので、退出命令が挿入され、減分値「1」が結合され、黒ランR17の始点「2」および終点「7」が順次に結合される。次い

で、y座標6の主走査線では、黒ラン数が直前のラインの黒ラン数と同数のなので、黒ランR18の始点「2」および終点「7」が順次に結合される。次いで、y座標8の主走査線は、白ラインなので、オフセット命令が挿入され、オフセット値「2」が結合される。

【0058】このように、具体例2の画像符号化方法によれば、副走査方向に1つ以上連続し、黒ラン数が同数の主走査線をグループ化し、主走査線内の黒ラン数が0の場合には、オフセット値をカウントし、オフセット命令にオフセット値を結合して符号を構成するとともに、主走査線内の黒ラン数が0でない場合には、主走査線の黒ラン数が直前の主走査線の黒ラン数より多いときには、進入命令に増分値および黒ランの位置情報を結合し符号を構成し、主走査線の黒ラン数が直前の主走査線の黒ラン数より少ないときには、退出命令に減分値および黒ランの位置情報を結合して符号を構成し、主走査線の黒ラン数が直前の主走査線の黒ラン数が同数のときにはそのまま黒ランの位置情報により符号を構成する。したがって、白画素の情報量を極めて少なくすることができるので、高い圧縮率の符号を得ることができる。また、複雑な演算を行うことなく2値画像を符号化することができるので、符号化処理を高速に行うことができる。さらに、符号化すべき主走査線と符号化済みの直前の主走査線の黒ラン数との差を符号化することで、黒ラン数を記述する符号長を短くすることができる。また、具体例2の画像復号化方法によれば、オフセット命令、進入命令および退出命令を解析し、符号化された2値画像を好適に復号化することができる。

【0059】〈画像符号化装置〉図8は本発明に係る具体例2の画像符号化装置の構成を示すブロック図である。図8に示すように、画像符号化装置は、画像メモリ31、黒ラン解析部32、黒ラン情報バッファ36、黒ラン数増減値算出部37、オフセット値カウンタ38、符号構成部39および符号メモリ40を備えている。

【0060】画像メモリ31には、符号化対象の2値画像が記憶されている。黒ラン解析部32は、画像メモリ11に記憶された2値画像を走査順に読み出し、読み出された走査線内の黒ランを解析するものであり、黒ラン始点解析部33、黒ラン長カウンタ34および黒ラン数カウンタ35を有する。黒ラン始点解析部33は、走査線内の黒ランの始点を解析し、解析された黒ランの始点を黒ラン情報バッファ36に出力する。黒ラン長カウンタ34は、黒ラン始点解析部33により黒ランの始点が解析された場合、その黒ランの黒ラン長をカウントし、カウントされた黒ラン長を黒ラン情報バッファ36に出力する。黒ラン数カウンタ35は、走査線内の黒ラン数をカウントし、カウントされた黒ラン数を黒ラン数増減値算出部37、オフセット値カウンタ38および符号構成部39に出力する。

【0061】黒ラン位置バッファ36は、黒ラン始点解

析部33により解析された黒ランの始点および黒ラン長カウンタ34によりカウントされた対応する黒ラン長を黒ラン情報として一時的に保持するバッファである。黒ラン数増減値算出部37は、前回、黒ラン数カウンタ35によりカウントされた直前の主走査線の黒ラン数を保持する図示しない保持部を有し、今回、黒ラン数カウンタ35によりカウントされた主走査線の黒ラン数から、前回、黒ラン数カウンタ35によりカウントされた主走査線の黒ラン数を減算し、減算値を符号構成部39に出力する。

【0062】オフセット値カウンタ38は、黒ラン数カウンタ35にカウントされた黒ラン数に基づいて、副走査方向に1つ以上連続する白ラインの数を表すオフセット値をカウントする。オフセット値カウンタ38は、入力された黒ラン数が0の場合には、オフセット値を1インクリメントし、入力された黒ラン数が0でない場合には、オフセット値を参照し、参照されたオフセット値が0でないとき、オフセット値を符号構成部39に出力する。

【0063】符号構成部39は、前回、黒ラン数カウンタ35によりカウントされた黒ラン数を保持する図示しない保持部を有し、今回、黒ラン数カウンタ35によりカウントされた黒ラン数に基づいて符号を構成する。符号構成部39は、今回、入力された黒ラン数が0でない場合には、前回、入力された黒ラン数を参照し、参照された前回の黒ラン数が0のとき、オフセット値カウンタ38により出力されたオフセット値の前にオフセット命令を挿入して符号を構成し、参照された前回の黒ラン数が0でないとき、黒ラン数増減値算出部37から出力された減算値を参照する。

【0064】符号構成部39は、黒ラン数増減値算出部37から出力された減算値が正值の場合には、今回、入力された減算値（増分値）の前に進入命令を挿入するとともに、黒ラン情報バッファ36に保持された黒ラン情報を読み出し、増分値に結合して符号を構成する。符号構成部39は、黒ラン数増減値算出部37から出力された減算値が負値の場合には、今回、入力された減算値の絶対値（減分値）の前に退出命令を挿入するとともに、黒ラン情報バッファ36に保持された黒ラン情報を読み出し、減分値に結合して符号を構成する。符号構成部39は、黒ラン数増減値算出部37から出力された減算値が0の場合には、黒ラン位置バッファ15に保持されたランの位置情報を読み出し、そのまま符号を構成する。符号構成部39により構成された符号は、符号メモリ40に記憶される。

【0065】図9および図10は図8に示された具体例2の画像符号化装置の動作を示すフローチャートである。まず、画像メモリ31に記憶された2値画像が走査順に1走査線毎に黒ラン解析部32に読み出され（ステップS31）、黒ラン始点解析部33により主走査線内

の黒ランの始点が解析されるとともに、黒ラン長カウンタ34により黒ラン長が解析され、黒ラン情報として黒ラン情報バッファ36に保持され、同時に、黒ラン数カウンタ35により主走査線内の黒ラン数がカウントされる(ステップS32)。

【0066】次いで、黒ラン数増減値算出部37により、今回、黒ラン数カウンタ35によりカウントされた主走査線の黒ラン数から、前回、黒ラン数カウンタ35によりカウントされた黒ラン数が減算され、減算値が符号構成部39に出力される(ステップS33)。次いで、オフセット値カウンタ38では、今回、黒ラン数カウンタ35によりカウントされた黒ラン数が0か否かが判定される(ステップS34)。ステップS34で、黒ラン数が0の場合には、オフセット値が1インクリメントされ(ステップS35)、ステップS46に進む。一方、ステップS34で、黒ラン数が0でない場合には、オフセット値が参照される(ステップS36)。ここで、オフセット値が0でない場合には、オフセット値が符号構成部39に出力され、オフセット値が0にリセットされ(ステップS37)、ステップS38に進む。一方、オフセット値が0の場合には、ステップS38に進む。

【0067】符号構成部39では、今回、黒ラン数カウンタ35によりカウントされた黒ラン数が0か否かが判定される(ステップS38)。ステップS38で、今回、黒ラン数カウンタ35によりカウントされた黒ラン数が0の場合には、ステップS46に進む。一方、ステップS38で、今回、黒ラン数カウンタ35によりカウントされた黒ラン数が0でない場合には、前回、黒ラン数カウンタ35によりカウントされた直前の主走査線の黒ラン数が0か否かが判定される(ステップS39)。ステップS39で、直前の主走査線の黒ラン数が0でない場合には、ステップS41に進む。一方、ステップS39で、直前の主走査線の黒ラン数が0のときには、オフセット値カウンタ38により出力されたオフセット値の前にオフセット命令が挿入され、符号が構成される(ステップS40)。

【0068】次いで、黒ラン数増減値算出部37から出力された減算値が正值であるか否かが判定される(ステップS41)。ステップS41で、減算値が正值の場合には、減算値(増分値)の前に進入命令が挿入され、黒ラン情報バッファ36に保持された黒ラン情報が読み出され、減算値に結合されて符号が構成される(ステップS42)。一方、ステップS41で、減算値が正值でない場合には、減算値が負値であるか否かが判定される(ステップS43)。ステップS43で、減算値が負値の場合には、減算値の絶対値(減分値)の前に退出命令が挿入され、黒ラン情報バッファ36に保持された黒ラン情報が読み出され、減算値に結合されて符号が構成される(ステップS44)。一方、ステップS43で、減

算値が負値でない場合、すなわち減算値が0の場合には、黒ラン情報バッファ36に保持された黒ラン情報が読み出され、そのまま符号が構成される(ステップS45)。

【0069】次いで、ステップS46において、次の主走査線の有無が判定され、次の主走査線がある場合には、ステップS31に戻る。一方、ステップS46で、次の走査線が無い場合には、オフセット値カウンタ38のオフセット値が0か否かが判定され(ステップS47)、オフセット値が0でない場合には、符号構成部39により、このオフセット値の前にオフセット命令が挿入されて符号が構成され、処理を終了する。一方、ステップS47で、オフセット値が0の場合には、処理を終了する。

【0070】このように、具体例2の画像符号化装置によれば、黒ラン数カウンタ35によりカウントされた黒ラン数が0の場合には、オフセット命令をカウントし、黒ラン数カウンタ35によりカウントされた黒ラン数が0でない場合には、オフセット値が0でないとき、オフセット命令にオフセット値を結合して符号を構成し、黒ラン数増減値算出部37の減算値が正值のとき、進入命令に増分値および黒ラン情報を結合して符号を構成し、黒ラン数増減値算出部37の減算値が負値のとき、退出命令に減分値および黒ラン情報を結合して符号を構成し、黒ラン数増減値算出部37の減算値が0のとき、黒ラン情報をそのまま符号化する。したがって、上記の具体例2の画像符号化方法を好適に実現することができ

る。

【0071】〈画像復号化装置〉図11は本発明に係る具体例2の画像復号化装置の構成を示すブロック図である。図11に示すように、画像復号化装置は、図8に示された画像符号化装置により符号化された2値画像を復号化するものであり、符号メモリ41、復号化部42および画像メモリ45を備えている。

【0072】符号メモリ41には、復号化対象の符号データが記憶されている。復号化部42は、符号メモリ41に記憶された符号データをその先頭から読み出し、復号化するものであり、命令解析部43および黒ラン数カウンタ44を有する。命令解析部43は、入力された符号データに含まれるオフセット命令、進入命令および退出命令を解析する。復号化部42は、命令解析部43によりオフセット命令が解析された場合には、後続するオフセット値を解析し、解析されたオフセット値に相当する白ラインを復号化する。復号化部42は、進入命令が解析された場合には、進入命令に後続する黒ラン数の増分値を解析し、解析された増分値を黒ラン数カウンタ44に出力する。復号化部42は、退出命令が解析された場合には、退出命令に後続する黒ラン数の減分値を解析し、解析された減分値に負値を表す符号を付加して黒ラン数カウンタ44に出力する。

【0073】黒ラン数カウンタ44は、復号化すべき主走査線の黒ラン数をカウントするものであり、初期値は0にセットされている。黒ラン数カウンタ44は、命令解析部23によりオフセット命令が解析された場合には、カウント値を0にリセットする。黒ラン数カウンタ44は、復号化部22により解析された増分値が入力された場合、入力された増分値の分だけカウント値をインクリメントして、復号化すべき黒ラン数を算出する。黒ラン数カウンタ44は、復号化部22により解析され、負値を表す符号が付加された減分値が入力された場合、入力された減分値の分だけカウント値をデクリメントして、復号化すべき黒ラン数を算出する。

【0074】復号化部42は、進入命令および退出命令が解析された場合、それぞれ黒ラン数カウンタ44にインクリメントおよびデクリメントされたカウント値を解析し、解析された黒ラン数毎に、黒ラン数の減算値（増分値および減分値）に後続する黒ラン情報（黒ランの始点および黒ラン長）を取り出し、主走査線を復号化する。画像メモリ45には、復号化部42により復号化された2値画像が記憶される。

【0075】図12は図11に示された具体例2の画像復号化装置の動作を示すフローチャートである。まず、符号メモリ41に記憶された符号データがその先頭から読み出され、復号化部42に入力される（ステップS51）。次いで、復号化部42の命令解析部43により入力された符号に含まれるオフセット命令が解析される（ステップS52）。ステップS52で、オフセット命令が解析された場合には、オフセット命令に後続するオフセット値が解析され、解析されたオフセット値に相当する白ラインが復号化され（ステップS53）、黒ラン数カウンタ44のカウント値が0にリセットされ（ステップS54）、ステップS62に進む。

【0076】一方、ステップS52において、オフセット命令が解析されない場合には、符号データに含まれる進入命令が解析される（ステップS55）。ステップS55において、進入命令が解析された場合には、進入命令に後続する黒ラン数の増分値が解析され、解析された増分値が黒ラン数カウンタ44に出力される（ステップS56）。次いで、黒ラン数カウンタ44により、入力された増分値の分、カウント値がインクリメントされる（ステップS57）。次いで、復号化部42により黒ラン数カウンタ44のカウント値が解析され、解析された黒ラン数の後続する黒ラン情報が取り出され、主走査線が復号化される（ステップS58）。

【0077】一方、ステップS55において、進入命令が解析されない場合には、符号データに含まれる退出命令が解析される（ステップS59）。ステップS59において、退出命令が解析された場合には、退出命令に後続する黒ラン数の減分値が解析され、解析された減分値に負値を表す符号が付加されて黒ラン数カウンタ44に

出力される（ステップS60）。次いで、黒ラン数カウンタ44により、入力された減分値の分、カウント値がデクリメントされ（ステップS61）、ステップS58に進む。

【0078】一方、ステップS59において、退出命令が解析されない場合には、ステップS58に進む。次いで、ステップS62において、次の符号データがある場合には、ステップS51に戻り、次の符号データがない場合には、処理を終了する。

【0079】このように、具体例2の画像復号化装置によれば、命令解析部43によりオフセット命令が解析された場合には、後続するオフセット値に相当する白ラインを復号化し、進入命令および退出命令が解析された場合には、黒ラン数カウンタ44により加算および減算された黒ラン数に基づいて主走査線を復号化し、何れの命令も解析されない場合には、現在の黒ラン数カウンタ44の黒ラン数に基づいて主走査線を復号化する。したがって、上記の具体例2の画像復号化方法を好適に実現することができる。

【0080】なお、上記具体例1および2では、2値画像を符号化するように構成しているが、2値画像のランレンクス符号を符号化するように構成してもよい。

【0081】《具体例3》

〈画像符号化装置〉図13は本発明に係る具体例1の画像符号化装置の構成を示すブロック図である。図13に示すように、画像符号化装置は、画像メモリ51、ラン解析部52、ラン情報バッファ55、ラン数比較部56、オフセット値カウンタ57、符号構成部58および符号メモリ59を備えている。

【0082】画像メモリ51には、白、黒、赤および青をそれぞれ表す複数の画素からなる4値画像が記憶されている。以下、白を表す画素を白画素、黒を表す画素を黒画素、赤を表す画素を赤画素、青を表す画素を青画素という。また、白画素のランを白ラン、黒画素のランを黒ラン、赤画素のランを赤ラン、青画素のランを青ランという。黒画素、赤画素および青画素はそれぞれ図形を表し、白画素はこれらにより表される図形の背景を表す。以下、全画素が白画素からなる主走査線を白ラインという。ラン解析部52は、画像メモリ51に記憶された4値画像を走査順に読み出し、読み出された走査線内の黒ラン、赤ランおよび青ランを解析するものであり、ラン位置解析部53およびラン総数カウンタ54を有する。ラン位置解析部53は、走査線内の画素の画素値に基づいて、黒ラン、赤ランおよび青ランのそれぞれの位置情報（始点および終点、または、始点およびラン長）を解析し、解析されたランの位置情報に当該ランの画素値を表す情報を付加してラン情報としてラン情報バッファ55に出力する。ラン総数カウンタ54は、走査線内の黒ラン、赤ランおよび青ランの総数をカウントし、カウントされたラン数をラン数比較部56、オフセット値

カウンタ57および符号構成部58に出力する。

【0083】ラン情報バッファ55は、ラン位置解析部53から出力された主走査線内の黒ラン、赤ラン、青ランのラン情報を一時的に保持するバッファである。ラン数比較部56は、前回、ラン総数カウンタ54によりカウントされた直前の主走査線のラン数を保持する図示しない保持部を有し、前回、ラン総数カウンタ54によりカウントされた主走査線のラン数と、今回、ラン総数カウンタ54によりカウントされた主走査線のラン数とを比較する。ラン数比較部56は、入力されたラン数と直前の主走査線のラン数とが一致した場合には、一致信号を符号構成部58に出力し、入力されたラン数と直前の主走査線のラン数とが一致しない場合には、不一致信号を符号構成部58に出力する。

【0084】オフセット値カウンタ57は、ラン総数カウンタ54によりカウントされたラン数に基づいて、副走査方向に1つ以上連続する白ラインの数を表すオフセット値をカウントする。オフセット値カウンタ57は、入力されたラン数が0の場合には、オフセット値を1インクリメントし、入力されたラン数が0でない場合には、オフセット値を参照し、参照されたオフセット値が0でないとき、オフセット値を符号構成部58に出力する。

【0085】符号構成部58は、前回、ラン総数カウンタ54によりカウントされたラン数を保持する図示しない保持部を有し、今回、ラン総数カウンタ54によりカウントされたラン数に基づいて符号を構成する。符号構成部58は、今回、入力されたラン数が0でない場合には、前回、入力されたラン数を参照し、参照された前回のラン数が0のとき、オフセット値カウンタ57により出力されたオフセット値の前にオフセット命令を挿入して符号を構成し、参照された前回のラン数が0でないとき、ラン数比較部56により出力された信号を参照する。

【0086】符号構成部58は、ラン数比較部56から出力された信号が不一致信号の場合には、今回、入力されたラン数の前にラン命令を挿入するとともに、ラン情報バッファ55に保持された黒ラン、赤ランおよび青ランのそれぞれのラン情報を読み出し、今回、入力されたラン数に結合して符号を構成する。符号構成部58は、ラン数比較部56から出力された信号が一致信号の場合には、ラン情報バッファ55に保持された黒ラン、赤ランおよび青ランのそれぞれのラン情報を読み出し、そのまま符号を構成する。符号構成部58により構成された符号は、符号メモリ59に記憶される。

【0087】すなわち、具体例1および2では、2値画像を符号化対象としたのに対し、具体例3の画像符号化装置は、黒ラン、赤ランおよび青ランのそれぞれのランの位置情報にそれぞれの画素値を付加することで、符号化対象を多値画像に拡張したものである。なお、具体例

3の画像符号化装置の動作は、黒ラン解析部12により黒ラン、赤ランおよび青ランのラン情報をそれぞれ解析する以外は、図3および図4に示された具体例1の画像符号化装置の動作と同様であり、その説明を省略する。

【0088】このように、具体例3の画像符号化装置によれば、副走査方向に1つ以上連続し、黒ラン、赤ランおよび青ランの総数が同数の主走査線をグループ化し、ランの総数が0の場合には、オフセット値をカウントし、オフセット命令にオフセット値を結合して符号を構成するとともに、ランの総数が0でない場合には、ラン命令にラン数および黒ラン、赤ランおよび青ランのそれぞれのラン情報（位置情報および画素値）を結合して符号を構成する。したがって、白画素の情報量を極めて少なくすることができるので、高い圧縮率の符号を得ることができる。また、複雑な演算を行うことなく2値画像を符号化することができるので、符号化処理を高速に行うことができる。

【0089】〈画像復号化装置〉図14は本発明に係る具体例3の画像復号化装置の構成を示すブロック図である。図14に示すように、画像復号化装置は、図13に示された画像符号化装置により符号化された4値画像を復号化するものであり、符号メモリ61、復号化部62および画像メモリ64を備えている。

【0090】符号メモリ61には、4値画像の符号データが記憶されている。復号化部62は、符号メモリ61に記憶された符号データをその先頭から読み出し、復号化するものであり、命令解析部63および復号化すべき主走査線のラン数を保持する図示しない保持部を有する。命令解析部63は、入力された符号データに含まれるオフセット命令およびラン命令を解析する。復号化部62は、命令解析部63によりオフセット命令が解析された場合には、後続するオフセット値を解析し、解析されたオフセット値に相当する白ラインを復号化する。復号化部62は、命令解析部63によりラン命令が解析された場合には、後続するラン数を解析して図示しない保持部に保持し、保持されたラン数毎に、ラン数に後続する黒ラン、赤ランまたは青ランのラン情報（ランの始点、終点および画素値）を取り出し、主走査線を復号化する。画像メモリ64には、復号化部22により復号化された2値画像が記憶される。なお、具体例3の画像復号化装置の動作は、ラン情報に含まれる画素値に基づいて黒ラン、赤ランおよび青ランをそれぞれ復号化する以外は、図6に示された具体例1の画像復号化装置の動作と同様であり、その説明を省略する。

【0091】このように、具体例3の画像復号化装置によれば、オフセット命令およびラン命令を解析し、符号化された4値画像を好適に復号化することができる。また、複雑な演算を行うことなく符号化された4値画像を復号化することができるので、復号化処理を高速に行うことができる。具体例3の画像符号化装置および画像復

号化装置は、特に、比較的色数の少ないカラー画像や比較的階調数が少ないグレースケール画像であって、図形を表す画素に対し、図形の背景を表す画素の割合が極めて多い画像に対して有効である。

【0092】《具体例4》

〈画像符号化装置〉図15は本発明に係る具体例1の画像符号化装置の構成を示すブロック図である。図15に示すように、画像符号化装置は、画像メモリ71、ラン解析部72、ラン情報バッファ75、ラン数比較部76、オフセット値カウンタ77、符号構成部78および符号メモリ79を備えている。

【0093】画像メモリ71には、8値のグレースケール画像が記憶されている。この8値のうち画素値が最大の画素は、図形の背景を表し、他の画素値の画素は、図形を表す。以下、画素値が最大の画素を白画素といい、全画素が白画素からなる主走査線を白ラインという。また、主走査線内で、白画素以外の画素が1つ以上連続する画素列をランという。ラン解析部72は、画像メモリ71に記憶された8値のグレースケール画像を走査順に読み出し、読み出された走査線内のランを解析するものであり、ラン位置解析部73およびラン総数カウンタ74を有する。ラン位置解析部73は、走査線内の画素の画素値に基づいて、ランの始点および終点を解析し、解析されたランの始点にその画素値を付加するとともに、ランの終点にその画素値を付加し、ラン情報としてラン情報バッファ75に出力する。ラン数カウンタ74は、走査線内のランの数をカウントし、カウントされたラン数をラン数比較部76、オフセット値カウンタ77および符号構成部78に出力する。

【0094】ラン情報バッファ75は、ラン位置解析部73から出力された主走査線内のラン情報を一時的に保持するバッファである。ラン数比較部76は、前回、ラン数カウンタ74によりカウントされた直前の主走査線のラン数を保持する図示しない保持部を有し、前回、ラン数カウンタ74によりカウントされた主走査線のラン数と、今回、ラン数カウンタ74によりカウントされた主走査線のラン数とを比較する。ラン数比較部76は、入力されたラン数と直前の主走査線のラン数とが一致した場合には、一致信号を符号構成部78に出力し、入力されたラン数と直前の主走査線のラン数とが一致しない場合には、不一致信号を符号構成部78に出力する。

【0095】オフセット値カウンタ77は、ラン数カウンタ74によりカウントされたラン数に基づいて、副走査方向に1つ以上連続する白ラインの数を表すオフセット値をカウントする。オフセット値カウンタ77は、入力されたラン数が0の場合には、オフセット値を1インクリメントし、入力されたラン数が0でない場合には、オフセット値を参照し、参照されたオフセット値が0でないとき、オフセット値を符号構成部78に出力する。

【0096】符号構成部78は、前回、ラン数カウンタ

74によりカウントされたラン数を保持する図示しない保持部を有し、今回、ラン数カウンタ74によりカウントされたラン数に基づいて符号を構成する。符号構成部78は、今回、入力されたラン数が0でない場合には、前回、入力されたラン数を参照し、参照された前回のラン数が0のとき、オフセット値カウンタ77により出力されたオフセット値の前にオフセット命令を挿入して符号を構成し、参照された前回のラン数が0でないとき、ラン数比較部76により出力された信号を参照する。

【0097】符号構成部78は、ラン数比較部76から出力された信号が不一致信号の場合には、今回、入力されたラン数の前にラン命令を挿入するとともに、ラン情報バッファ75に保持されたラン情報を読み出し、今回、入力されたラン数に結合して符号を構成する。符号構成部78は、ラン数比較部76から出力された信号が一致信号の場合には、ラン情報バッファ75に保持されたラン情報を読み出し、そのまま符号を構成する。符号構成部78により構成された符号は、符号メモリ79に記憶される。

【0098】すなわち、具体例4の画像符号化装置は、図形の背景を表す白画素以外の画素からなるランの始点および終点にそれぞれの画素値を付加し、多値画像を符号化するものである。なお、具体例4の画像符号化装置の動作は、ラン解析部73によりラン情報を解析する以外は、図3および図4に示された具体例1の画像符号化装置の動作と同様であり、その説明を省略する。

【0099】このように、具体例4の画像符号化装置によれば、副走査方向に1つ以上連続し、白画素以外の画素からなるラン数が同数の主走査線をグループ化し、ランの数が0の場合には、オフセット値をカウントし、オフセット命令にオフセット値を結合して符号を構成するとともに、ラン数が0でない場合には、ラン命令にラン数およびラン情報（ランの始点およびその画素値、並びに、ランの終点およびその画素値）を結合して符号を構成する。したがって、白画素の情報量を極めて少なくすることができるので、高い圧縮率の符号を得ることができる。また、複雑な演算を行うことなく2値画像を符号化することができるので、符号化処理を高速に行うことができる。

【0100】〈画像復号化装置〉図16は本発明に係る具体例4の画像復号化装置の構成を示すブロック図である。図16に示すように、画像復号化装置は、図15に示された画像符号化装置により符号化された8値のグレースケール画像を復号化するものであり、符号メモリ81、復号化部82、画素補間部84および画像メモリ85を備えている。

【0101】符号メモリ81には、8値のグレースケール画像の符号データが記憶されている。復号化部82は、符号メモリ81に記憶された符号データをその先頭から読み出し、復号化するものであり、命令解析部83



および復号化すべき主走査線のラン数を保持する図示しない保持部を有する。命令解析部83は、入力された符号データに含まれるオフセット命令およびラン命令を解析する。復号化部82は、命令解析部83によりオフセット命令が解析された場合には、後続するオフセット値を解析し、解析されたオフセット値に相当する白ラインを復号化する。復号化部82は、命令解析部83によりラン命令が解析された場合には、後続するラン数を解析して図示しない保持部に保持し、保持されたラン数毎に、ラン数に後続するラン情報(ランの始点およびその画素値、ランの終点およびその画素値)を取り出し、画素補間部84に出力する。

【0102】画素補間部84は、入力されたラン情報に基づいてランの始点および終点間の画素を補間する。画素補間部84は、始点および終点に基づいてラン長を求め、終点の画素値から始点の画素値を減算し、減算値を(ラン長-1)で除して、始点から終点に向かう1画素当たりの画素値の増分値を算出する。そして、画素補間部84は、ランの始点に増分値を加算していき、始点および終点間の画素を近似的に復元する。復元された画素は白画素以外の7値の何れかに近似される。復号化部82は、画素補間部84により復元されたランを白ライン上に配置して主走査線を復号化する。画像メモリ85には、復号化部22により復号化された8値のグレースケール画像が記憶される。なお、具体例4の画像復号化装置の動作は、画素補間部84によりラン情報に基づいてランが復元され、復号化部82により主走査線が復号化される以外は、図6に示された具体例1の画像復号化装置の動作と同様であり、その説明を省略する。

【0103】このように、具体例4の画像復号化装置によれば、オフセット命令およびラン命令を解析し、符号化されたグレースケール画像を好適に復号化することができる。具体例4の画像符号化装置および画像復号化装置は、特に、比較的の階調数が少ないグレースケール画像や3次元的な深さを画素値とする画像であって、図形を表す画素に対し、図形の背景を表す画素の割合が極めて多い画像に対して有効である。

【0104】なお、具体例4の画像符号化装置および画像復号化装置では、画像符号化装置によりランの始点およびその画素値、並びに、終点およびその画素値を符号化し、画像復号化装置により始点からの増分値を算出してランを復号するようにしているが、例えば、画像符号化装置により始点からの増分値を求め、始点およびその画素値、並びに、ラン長とともに符号を構成し、画像復号化装置では、始点の画素値に増分値を加算することによりランを復元するように構成してもよい。

【0105】なお、上記各具体例の画像符号化装置および画像復号化装置の動作をそれぞれ実行するプログラムを作成し、フレキシブルディスク、ハードディスク、ROM等の記録媒体に記録することができる。この場合、

装置を安価に提供することができる。本発明に係る画像符号化方法および画像復号化方法は、図形処理された画像をコンピュータにより符号化してプリンタに転送し、プリンタにより復号化して印刷するのに好適である。

【図面の簡単な説明】

【図1】本発明に係る具体例1の画像符号化方法の概略説明図であり、図1(a)は符号化対象の2値画像の一例を示す図、図1(b)は図1(a)に示された2値画像の符号構成を示す図である。

【図2】本発明に係る具体例1の画像符号化装置の構成を示すブロック図である。

【図3】具体例1の画像符号化装置の動作を示すフローチャート(その1)である。

【図4】具体例1の画像符号化装置の動作を示すフローチャート(その2)である。

【図5】本発明に係る具体例1の画像復号化装置の構成を示すブロック図である。

【図6】具体例1の画像復号化装置の動作を示すフローチャートである。

【図7】本発明に係る具体例2の画像符号化方法の概略説明図であり、図7(a)は符号化対象の2値画像の一例を示す図、図7(b)は図1(a)に示された2値画像の符号構成を示す図である。

【図8】本発明に係る具体例2の画像符号化装置の構成を示すブロック図である。

【図9】具体例2の画像符号化装置の動作を示すフローチャート(その1)である。

【図10】具体例2の画像符号化装置の動作を示すフローチャート(その2)である。

【図11】本発明に係る具体例2の画像復号化装置の構成を示すブロック図である。

【図12】具体例2の画像復号化装置の動作を示すフローチャートである。

【図13】本発明に係る具体例3の画像符号化装置の構成を示すブロック図である。

【図14】本発明に係る具体例3の画像復号化装置の構成を示すブロック図である。

【図15】本発明に係る具体例4の画像符号化装置の構成を示すブロック図である。

【図16】本発明に係る具体例4の画像復号化装置の構成を示すブロック図である。

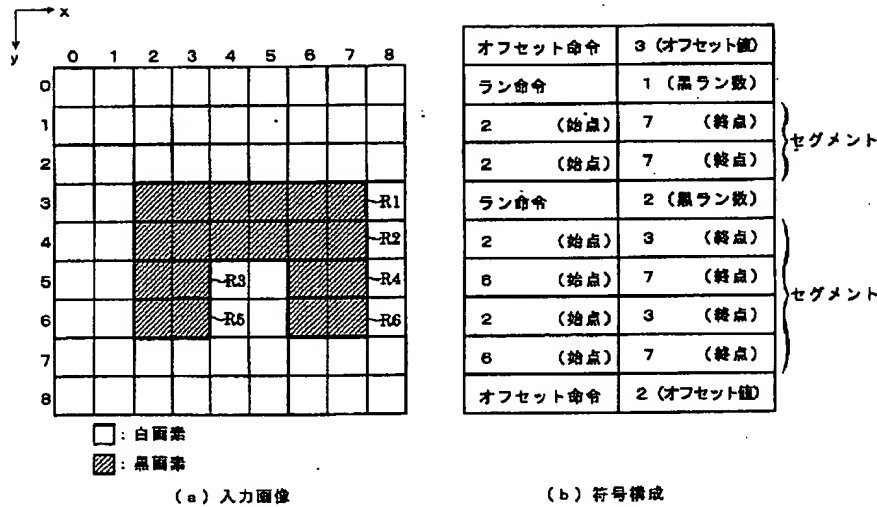
【符号の説明】

- 11 画像メモリ
- 12 黒ラン解析部
- 13 黒ラン位置解析部
- 14 黒ラン数カウンタ
- 15 黒ラン位置バッファ
- 16 黒ラン比較部
- 17 オフセット値カウンタ
- 18 符号構成部



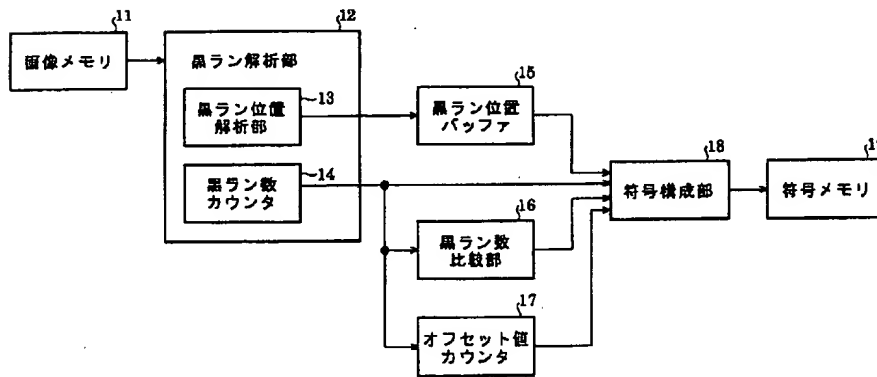
## 19 符号メモリ

【図1】



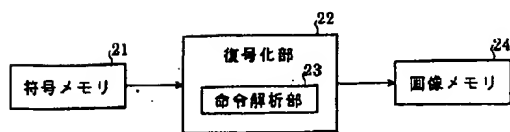
具体例1の画像符号化方法の概略説明図

【図2】



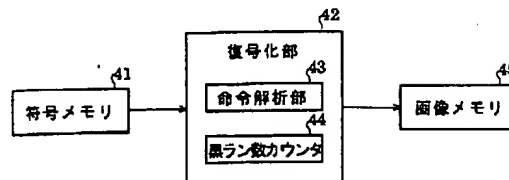
具体例1の画像符号化装置の構成を示すブロック図

【図5】



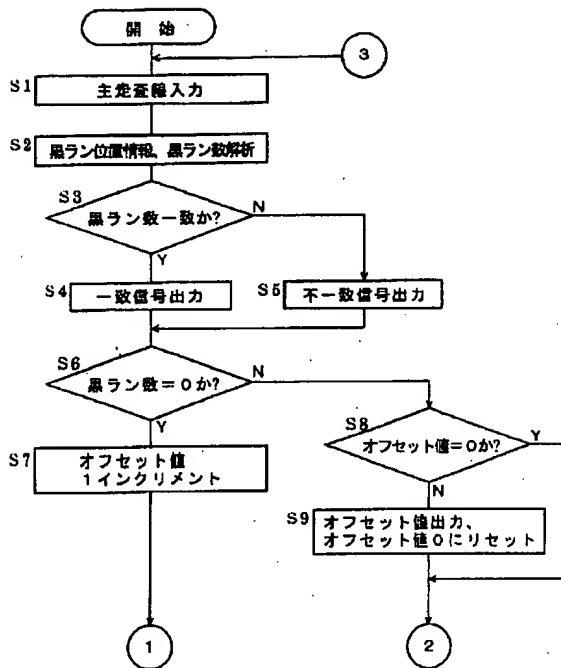
具体例1の画像復号化装置の構成を示すブロック図

【図11】



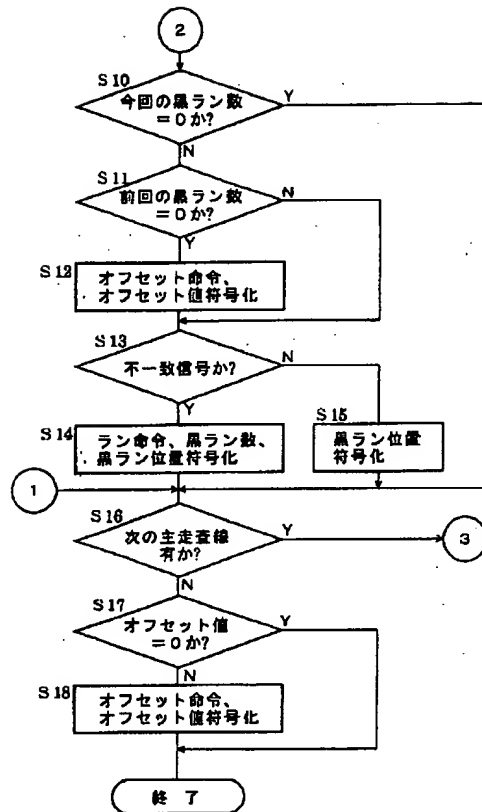
具体例2の画像復号化装置の構成を示すブロック図

【図3】



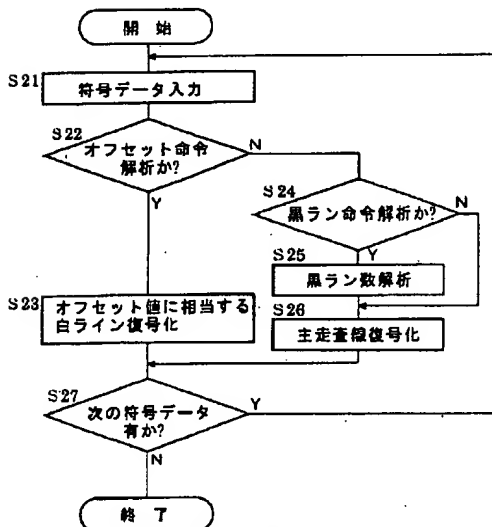
具体例1の画像符号化装置の動作を示すフローチャート（その1）

【図4】



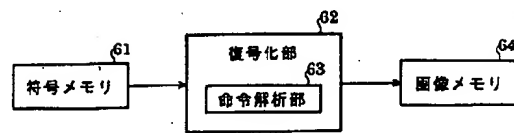
具体例1の画像符号化装置の動作を示すフローチャート（その2）

【図6】



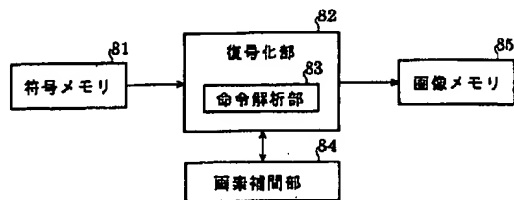
具体例1の画像復号化装置の動作を示すフローチャート

【図14】



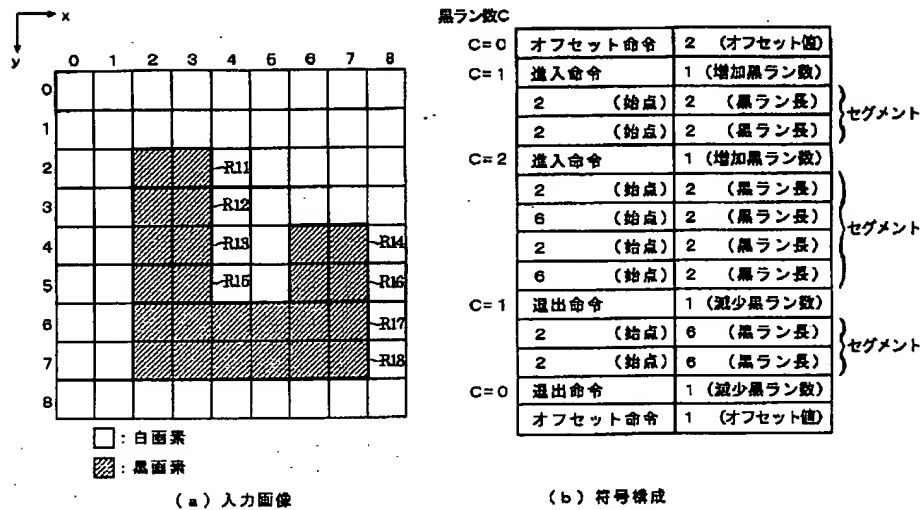
具体例3の画像復号化装置の構成を示すブロック図

【図16】



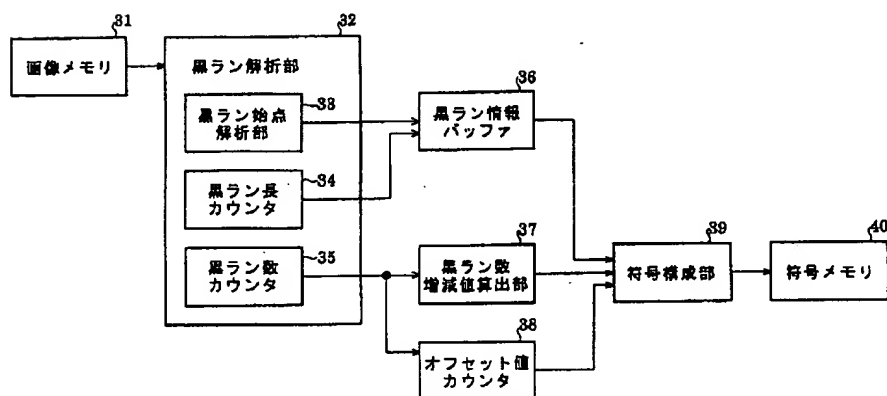
具体例4の画像復号化装置の構成を示すブロック図

【図7】



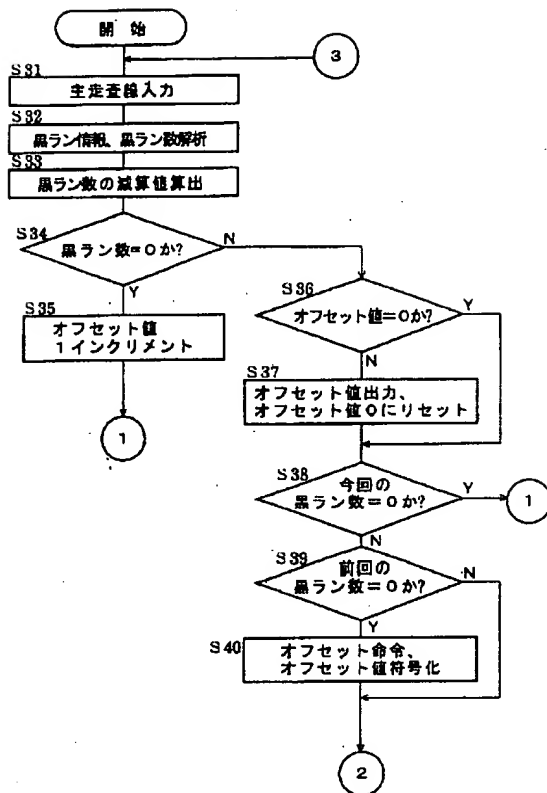
具体例2の画像符号化方法の概略説明図

【図8】



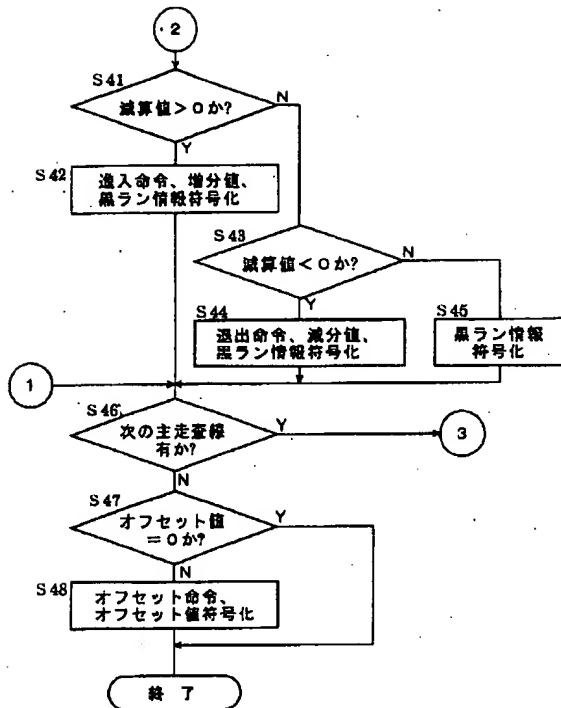
具体例2の画像符号化装置の構成を示すブロック図

【図9】



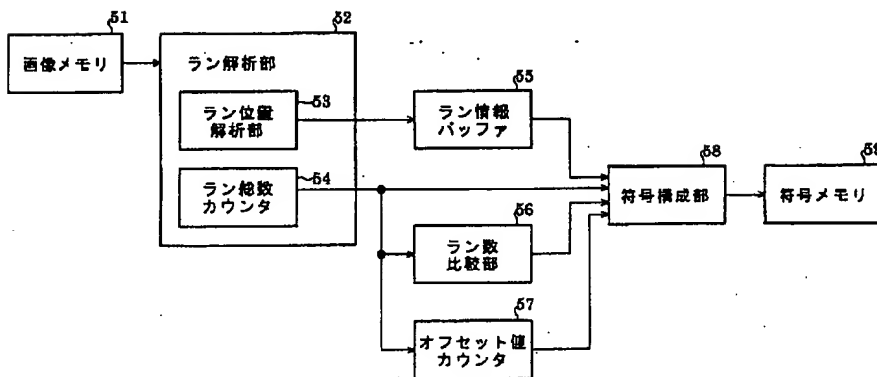
具体例 2 の画像符号化装置の動作を示すフローチャート（その 1）

【図10】



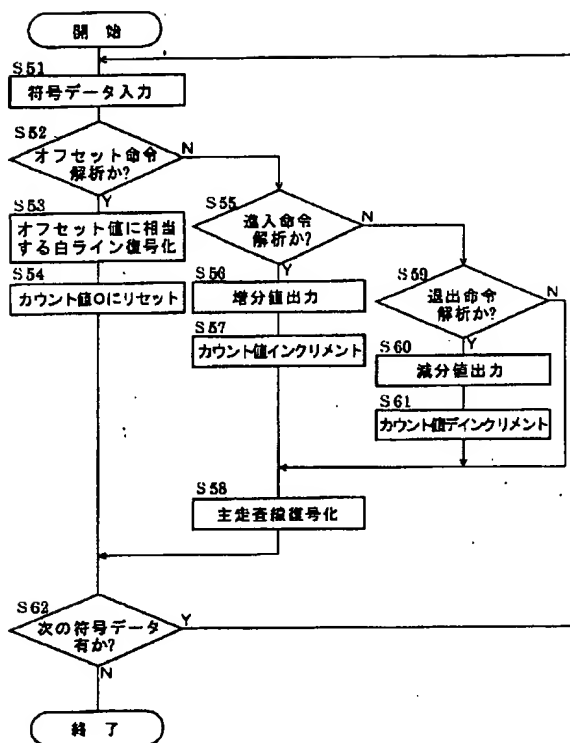
具体例 2 の画像符号化装置の動作を示すフローチャート (その 2)

【图 13】



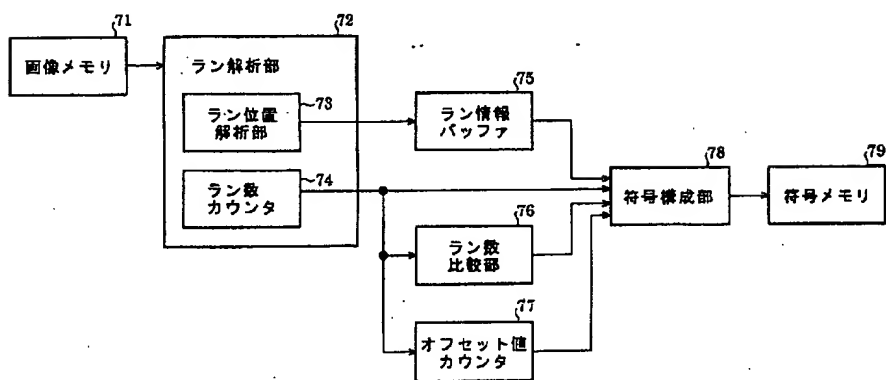
具体例 3 の画像符号化装置の構成を示すブロック図

【図12】



具体例2の画像復号化装置の動作を示すフローチャート

【図15】




具体例4の画像符号化装置の構成を示すブロック図

**IMAGE CODING/DECODING METHOD AND ITS SYSTEM**

**Patent number:** JP11027540  
**Publication date:** 1999-01-29  
**Inventor:** MATSUSHIRO NOBUHITO  
**Applicant:** OKI DATA:KK  
**Classification:**  
- international: H04N1/419  
- european:  
**Application number:** JP19970194807 19970704  
**Priority number(s):**

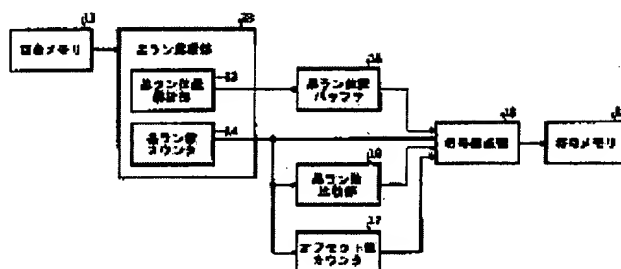
**Also published as:**

 US6301300 (B1)

**Abstract of JP11027540**

**PROBLEM TO BE SOLVED:** To enhance the compression rate by grouping main scanning lines which have same number of black runs, combining offset values to an offset instruction in the case that the black run number is 0, combining the run information to a run instruction when the run number is not 0, so as to configure a code, thereby reducing an information amount of white pixels.

**SOLUTION:** A binary image of an image memory 11 is read by a black run analysis section 12, a black run position analysis section 13 analyzes the position information of black runs in a main scanning line and the result is stored in a black run buffer 15, and a black run number counter 14 counts the number of black runs. When the black run number is zero, an offset counter counts an offset instruction. When the black run number is not 0 and an offset value is not 0, the offset value is combined with the offset instruction for configuring the code. Furthermore, a black run number comparison section 16 compares the black run number of a main scanning line with a black run number of an immediately preceding scanning line, and when they disagree, the black run number and the position information are combined to the run instruction to configure the code, and when they agree, the black run position information is coded as it is.



Data supplied from the **esp@cenet** database - Worldwide

## IMAGE CODING/DECODING METHOD AND ITS SYSTEM

Description of correspondent: **US6301300**

### BACKGROUND OF THE INVENTION

The present invention relates to a method and apparatus useful for coding and decoding images such as line drawings that comprise mostly background area.

Black-and-white bi-level images can be encoded by run-length coding, which encodes the number of consecutive picture elements or pixels having the same value. When this type of image is transmitted by facsimile, for example, the image is scanned a line at a time, and each scanning line is encoded as run lengths specifying the relative positions of the pixels at which a change from black to white, or from white to black, occurs. In the modified Huffman (MH) code adopted in group-three facsimile equipment, the run-length information is further coded by use of variable-length coding tables, using different tables for black runs and white runs, because the two types of runs have different statistical properties.

When this type of run-length coding is applied to an image comprising mostly white background area, however, most of the coded data is used to encode long runs of background pixels. Variable-length coding of the background in this type of image is inefficient, because the variable-length coding rule forces the use of many redundant bits. If the redundancy could be reduced, higher data compression ratios could be obtained.

More efficient coding schemes, such as arithmetic coding, are known, but they tend to be computationally demanding in terms of program complexity, memory usage, and processing time.

The above problems are not limited to bi-level images, but also occurs in other types of sparse images, including images that employ several colors, or multiple shades of gray. These problems moreover occur not only in facsimile transmission, but also in image storage and retrieval, and in the transmission of image data from a computer to a printer.

### SUMMARY OF THE INVENTION

An object of the present invention is to code sparse images efficiently, by reducing the amount of coded image data representing background areas.

Another object of the invention is to code and decode sparse images without extensive computation.

Another object is to code and decode sparse images without extensive memory usage.

The invented image coding method comprises the steps of:  
counting runs of non-background pixels, obtaining a run count for each scanning line in the image;  
counting consecutive scanning lines having run counts of zero;  
coding consecutive scanning lines having run counts of zero as a first command and a first value, the first value indicating the number of scanning lines; and  
coding consecutive scanning lines having identical run counts not equal to zero as a second command, a second value indicating the run count, and one-dimensional run information describing each run in each scanning line.

The second value may specify the run count directly, or may specify a value to be added to or subtracted from the previous run count. The run information may give the starting and ending coordinates of each run, or a starting coordinate and run length. If necessary, the run information may also give a single pixel value shared by all pixels in the run, a starting gray level and an ending gray level, or a starting gray level and a gray-level increment value.

The invented decoding method comprises the steps of:  
analyzing the coded image data to detect the first and second commands;  
generating entire scanning lines of background pixels according to the first command and its



accompanying first value; and  
generating scanning lines with runs of non-background pixels according to the second command, its accompanying second value, and the run information.

If necessary, the decoding method also comprises the steps of setting and modifying an internal run count according to the first command, second command, and second value, the step of assigning gray levels to the pixels in each run by interpolation between starting and ending gray levels, or the step of assigning gray levels by successively adding a gray-level increment value to a starting gray level.

The invented coder comprises a run analyzer for counting runs of non-background pixels and obtaining one-dimensional run information, a run information buffer for temporarily storing the run information for each scanning line, a run count comparator for comparing the run count of each scanning line with the run count of the preceding scanning line, an offset counter for counting consecutive scanning lines having run counts of zero, and a code assembler. The code assembler outputs a first command and first value when the run count changes from zero to a nonzero value, outputs a second command and second value when the run count changes from any value to a non-zero value, and outputs the run information stored in the run information buffer.

The invented decoder comprises a command analyzer for detecting the first and second commands, and a decoding section for generating scanning lines according to the first and second commands, first and second values, and run information. If necessary, the decoder also comprises a run counter and/or an interpolator.

#### BRIEF DESCRIPTION OF THE DRAWINGS

In the attached drawings:

FIG. 1A shows an example of a bi-level image;

FIG. 1B indicates how the image in FIG. 1A is coded according to a first embodiment of the invention;

FIG. 2 is a block diagram of a coder according to the first embodiment;

FIG. 3 is a first part of a flowchart illustrating coding operations in the first embodiment;

FIG. 4 is a second part of the flowchart in FIG. 3;

FIG. 5 is a block diagram of a decoder according to the first embodiment;

FIG. 6 is a flowchart illustrating decoding operations in the first embodiment;

FIG. 7A shows another example of a bi-level image;

FIG. 7B indicates how the image in FIG. 7A is coded according to a second embodiment of the invention;

FIG. 8 is a block diagram of a coder according to the second embodiment;

FIG. 9 is a first part of a flowchart illustrating coding operations in the second embodiment;

FIG. 10 is a second part of the flowchart in FIG. 9;

FIG. 11 is a block diagram of a decoder according to the second embodiment;

FIG. 12 is a flowchart illustrating decoding operations in the second embodiment;

FIG. 13 is a block diagram of a coder according to a third embodiment of the invention;

FIG. 14 is a block diagram of a decoder according to the third embodiment; FIG. 15 is a block diagram of a coder according to a fourth embodiment of the invention; and

FIG. 16 is a block diagram of a decoder according to the fourth embodiment.

## DETAILED DESCRIPTION OF THE INVENTION

Embodiments of the invention will be described with reference to the attached illustrative drawings.

FIG. 1A shows a sample image that will be used to explain the first embodiment of the present invention. The image is a bi-level image comprising only black and white pixels, the white pixels being background pixels. Cross-hatching is used to indicate black pixels.

The image has nine scanning lines of nine pixels each. In the indicated x-y coordinate system, scanning lines are identified by y-coordinates from zero to eight, and pixel positions in each scanning line by x-coordinates from zero to eight. The scanning lines are horizontal in the drawing. A scanning sequence is defined by taking the scanning lines in ascending order of their y-coordinates, and the pixels in each scanning line in ascending order of their x-coordinates.

A black run refers to a sequence of consecutive black pixels in the same scanning line. In FIG. 1A there are six black runs, numbered from R1 to R6.

The coding method of the first embodiment is based on counting the number of black runs in each scanning line. This number is referred to as the black run count of the scanning line. When all-white scanning lines occur, the number of consecutive all-white scanning lines, referred to below simply as white lines, is also counted; this number is referred to as an offset value. A group of one or more consecutive scanning lines having the same black run count, not immediately preceded or followed by a scanning line having the same black run count, is referred to as a segment.

With this terminology, the coding method of the first embodiment can be described as follows. A segment of white lines is coded as a single offset command followed by the offset value of the segment. A segment having a non-zero black run count is coded by a single run command, followed by the black run count, then the run information for each black run in each scanning line in the segment, taken in the scanning sequence. The run information is one-dimensional, giving the x-coordinates of the starting and ending pixels of the run, but not specifying the y-coordinate of the scanning line.

FIG. 1B indicates the coding of the image in FIG. 1A. The initial segment of three white lines is coded as an offset command, followed by the offset value of three, in the top line of FIG. 1B. The offset command and value are followed by a run command, indicated as 'runs/line' in FIG. 1B, and a black run count value of one. Run R1 is then coded by its starting x-coordinate (two) and ending x-coordinate (seven). Run R2 is coded similarly.

In the next two scanning lines, the black run count changes from one to two, so another run command is inserted, followed by a run count of two. Run R3 is coded by its starting x-coordinate (two) and ending x-coordinate (three). Run R4 is coded by its starting x-coordinate (six) and ending x-coordinate (seven). Runs R5 and R6 are coded similarly. The image ends with two white lines, which are coded by another offset command and offset value (two).

In an image of this size, the run coordinates, black run count values, and offset values can be coded as four-bit binary numbers taking on values from zero to nine. The run command and offset command can then be coded as four-bit binary numbers with values higher than nine: for example, 'fourteen' can designate a run command, and 'fifteen' an offset command. Each line in FIG. 1B then becomes a single eight-bit byte.

Images of arbitrary size can be coded in the same way by increasing the number of bits, the run and offset commands again being assigned values that are not used for run coordinates, black run counts, and offset values.

FIG. 2 shows an example of a coder employing the coding method of the first embodiment. The coder comprises an image memory 11, a black run analyzer 12 having a black run position analyzer 13 and black run counter 14, a black run information buffer 15, a black run count comparator 16, an offset counter 17, a code assembler 18, and a code memory 19.

The image memory 11 stores the image to be coded. The image may be stored in bit-mapped form, or in a compressed form such as a conventional run-length coded form.

The black run analyzer 12 reads the image data stored in the image memory 11 one scanning line at a time. The black run position analyzer 13 determines the starting and ending x-coordinates of each black run in the scanning line, and writes this run information in the black run information buffer 15. The black run counter 14 counts the number of black runs in the scanning line, and sends this information to the black run count comparator 16, offset counter 17, and code assembler 18.

The black run information buffer 15 temporarily stores the run information obtained by the black run position analyzer 13 for each scanning line, until the run information is read by the code assembler 18.

The black run count comparator 16 internally stores the run counts obtained by the black run counter 14 for the current scanning line and the preceding scanning line, compares these two counts, and sends the code assembler 18 a match signal or an unmatched signal, indicating whether the two counts agree.

The offset counter 17 maintains an internal offset value, which is incremented by one each time the black run counter 14 outputs a black run count of zero. At certain times, the offset counter 17 sends the offset value to the code assembler 18, and resets the offset value to zero.

The code assembler 18 writes offset commands, offset values, run commands, black run count values, and run information as coded data in the code memory 19. The offset values, black run count values, and run information are preferably written as binary numbers, without variable-length coding. The code assembler 18 also maintains an internal flag that indicates whether the black run count in the preceding scanning line was zero.

The code memory 19 stores the coded data pending transmission over a communication channel or other use.

The operation of the coder in FIG. 2 will now be described in more detail with reference to FIGS. 3 and 4. These flowcharts illustrate the processing of one scanning line.

Referring to FIG. 3, in step S1, the black run analyzer 12 reads one scanning line from the image memory 11. In step S2, the black run counter 14 counts the number of black runs in the scanning line, and the black run position analyzer 13 writes the starting and ending x-coordinates of each black run in the black run information buffer 15.

In step S3, the black run count comparator 16 compares the black run count of the current scanning line with the black run count of the preceding scanning line. If both scanning lines have the same black run count, the black run count comparator 16 outputs a match signal in step S4. If the two counts differ, the black run count comparator 16 outputs an unmatched signal in step S5.

In step S6, the offset counter 17 tests the black run count obtained by the black run counter 14 for the current scanning line. If the black run count is zero, then in step S7, the offset counter 17 internally increments the offset value, after which processing proceeds to step S16 in FIG. 4. If the black run count is not zero, then in step S8, the offset counter 17 internally tests the offset value. If the offset value is not zero, then in step S9, the offset counter 17 sends the offset value to the code assembler 18, and internally resets the offset value to zero. After step S9, or if the offset value is zero in step S8, processing proceeds to step S10 in FIG. 4.

In step S10, the code assembler 18 tests the black run count output by the black run counter 14 for the current scanning line. If the black run count is zero, processing branches to step S16, described below.

If the black run count is not zero, then in step S11, the code assembler 18 tests its internal flag to determine whether the black run count in the preceding scanning line was zero. If so, then in step S12, the code assembler 18 writes an offset command in the code memory 19, followed by the non-zero offset value received from the offset counter 17.

Following step S12, or if the black run count of the preceding scanning line was not zero in step S11, in step S13 the code assembler 18 tests the signal received from the black run count comparator 16. If an unmatched signal was received, then in step S14, the code assembler 18 writes a run command in the code memory 19, followed by the black run count received from the black run counter 14, and the run information read from the black run information buffer 15 for all of the black runs in the current scanning

line. If a match signal was received, then in step S15, the code assembler 18 simply writes the run information read from the black run information buffer 15 in the code memory 19.

Following step S7, step S14, step S15, or an affirmative result in step S10, a decision is made in step S16 as to whether there are any more scanning lines to be coded. This decision can be made on the basis of, for example, an end-of-file code stored in the image memory 11. If another scanning line is present, the process returns to step S1 in FIG. 3. If no further scanning lines are present, then in step S17, the offset counter 17 tests the current offset value. If this value is zero, the coding process ends. If the offset value is not zero, then in step S18 the code assembler 18 writes an offset command and the offset value in the code memory 19, after which the coding process ends.

By following the procedure in FIGS. 3 and 4, the code assembler 18 writes the starting and ending x-coordinates of all black runs in the code memory 19, writes a run command and black run count when the black run count per scanning line changes from any value to a non-zero value, and writes an offset command and offset value when the black run count changes from zero to a non-zero value. This procedure produces coded data of the type shown in FIG. 1B.

FIG. 5 shows the configuration of a decoder for decoding the coded image data produced by the coder in FIG. 2. The decoder comprises a code memory 21, a decoding section 22 having a command analyzer 23, and an image memory 24.

The code memory 21 stores the coded image data. The decoding section 22 reads the coded image data in sequence from beginning to end, stores the most recently read black run count, generates decoded image data, and writes the decoded image data in the image memory 24. The command analyzer 23 analyzes commands occurring in the coded image data. The image memory 24 stores the decoded image data pending printing or other use.

The operation of the decoder in FIG. 5 will be described with reference to FIG. 6.

In step S21, the decoding section 22 reads data from the code memory 21. The amount of data read corresponds, for example, to the length of one offset or run command and the accompanying offset or run-count value, or the length of a pair of x-coordinates.

In step S22, the command analyzer 23 attempts to detect an offset command in the data read in step S21. If an offset command is detected, in step S22 the decoding section 22 generates the number of white lines specified by the offset value that follows the offset command, and writes these lines into the image memory 24.

If an offset command is not detected, then in step S24, the command analyzer 23 attempts to detect a run command. If a run command is detected, then in step S25, the decoding section 22 internally stores the black run count that follows the run command.

Following step S25, or if no run command is detected in step S24, the decoding section 22 reads further data from the code memory 21 as necessary to obtain run information, generates a decoded scanning line containing black runs starting and ending at the positions specified by the run information, and writes this scanning line in the image memory 24. The amount of data to be read in step S26 is determined from the black run count most recently obtained in step S25.

Following step S23 or S26, in step S27 the decoding section 22 decides whether the code memory 21 contains any more data to be decoded. If so, processing returns to step S21 to begin decoding the next scanning line. If not, the decoding process ends.

As described above, the first embodiment enables an arbitrary number of white lines to be coded by a single offset command and value, and uses a single run command to designate a segment of an arbitrary number of consecutive lines with identical non-zero black run counts. For line drawings and other sparse images comprising mostly background, this coding scheme is more efficient than conventional schemes such as the coding scheme used in group-three facsimile transmissions. A particular advantage is that no special code is required to indicate the end of each scanning line.

The first embodiment is also computationally simple, and does not require the storage of large variable-length coding tables or large amounts of temporary data. These advantages enable the coding and

decoding processes to be carried out at high speed, using only a small amount of memory.

FIG. 7A shows a sample image that will be used to explain a second embodiment of the present invention. The image is generally similar to the image in FIG. 1A, but contains eight black runs, numbered from R11 to R18.

The second embodiment differs from the first embodiment in regard to the coding of run information. Instead of coding the starting and ending positions of each run, the second embodiment codes the starting position and run length. The second embodiment also replaces the run command of the first embodiment with a pair of commands that alter the black run count: an add command increases the black run count; a subtract command decreases the black run count. The add and subtract commands are followed by the value to be added to or subtracted from the run count. The second embodiment employs the same offset command and offset value as in the first embodiment.

FIG. 7B illustrates the coding of the image in FIG. 7A according to the second embodiment, and indicates the value of the black run count C at each point in the coded data.

The two white lines at the top of FIG. 7A are coded at the top of FIG. 7B as an offset command with an offset value of two. The offset command resets the black run count value C to zero, as indicated.

After these two white lines, the black run count per scanning line increases from zero to one, so an add command is inserted, followed by a value of one. The next two lines in FIG. 7B encode runs R11 and R12, giving the starting x-coordinate (two) and length (two) of each run.

The black run count per scanning line now increases again, from one to two, so another add command is inserted, followed by a value of one. The next four lines in FIG. 7B encode runs R13, R14, R15, and R16.

Next, the black run count per scanning line decreases from two to one, so a subtract command is inserted, followed by a value of one. The next two lines in FIG. 7B encode runs R17 and R18.

The final white line in FIG. 1A is coded in FIG. 1B as another offset command, with an offset value of one.

FIG. 8 shows an example of a coder employing the coding method of the second embodiment. The coder comprises an image memory 31, a black run analyzer 32 having a black run starting position analyzer 33, black run length counter 34, and black run counter 35, a black run information buffer 36, a black run count comparator 37, an offset counter 38, a code assembler 39, and a code memory 40.

The image memory 31 is similar to the image memory 11 in the first embodiment.

The black run analyzer 32 reads the image data stored in the image memory 31 one scanning line at a time. The black run starting position analyzer 33 and black run length counter 34 analyze the black runs in the scanning line. The black run starting position analyzer 33 stores the x-coordinate of the starting position of each black run in the black run information buffer 36. The black run length counter 34 stores the run length of each black run in the black run information buffer 36. The black run counter 35 counts the number of black runs in the scanning line, and sends the black run count value for each scanning line to the black run count comparator 37 and the offset counter 38.

The black run information buffer 36 temporarily stores the run information obtained by the black run start position analyzer 33 and black run length counter 34 for each scanning line, until read by the code assembler 39.

The black run count comparator 37 stores the black run counts output by the black run counter 14 for the current scanning line and the preceding line in registers (not visible), subtracts the black run count of the preceding scanning line from the black run count of the current scanning line, and sends the difference to the code assembler 39.

The offset counter 38 is similar to the offset counter 17 in the first embodiment.

The code assembler 39 writes offset commands, offset values, add commands, subtract commands, absolute difference values, and run information in the code memory 40, according to information received from the black run information buffer 36, black run count comparator 37, and offset counter 38. The code

assembler 39 also has an internal flag that indicates whether the black run count in the preceding scanning line was zero.

The code memory 40 is similar to the code memory 19 in the first embodiment.

The operation of the coder in FIG. 8 will now be described in more detail with reference to FIGS. 9 and 10.

Steps S31, S32, and S33 are generally similar to steps S<sub>i</sub> and S2 in the first embodiment. One scanning line is read and analyzed to obtain its run information and black run count. The run information, now comprising the starting x-coordinate and length of each black run, is stored in the black run information buffer 36.

Steps S34, S35, S36, and S37 are similar to steps S6, S7, S8, and S9 in the first embodiment. The offset counter 38 tests the black run count of the current scanning line, and increments an internal offset value if the black run count is zero. If the black run count is not zero and the offset value maintained in the offset counter 38 is also not zero, the offset counter 38 sends the offset value to the code assembler 39, then internally resets the offset value to zero. Processing then proceeds to step S46 in FIG. 10 if the black run count of the current scanning line is zero, or to step S38 if the black run count is not zero.

In step S38, the black run count comparator 37 also tests the black run count of the current scanning line. If the black run count is zero, processing proceeds to step S46 in FIG. 10. If the black run count is not zero, then the black run count comparator 37 subtracts the black run count of the preceding scanning line from the black run count of the current scanning line, and sends the difference to the code assembler 39.

In step S39, the code assembler 39 tests its internal flag to determine whether the black run count of the preceding scanning line was zero. If so, then in step S40, the code assembler 39 writes an offset command and the offset value received from the offset counter 38 in the code memory 40. Following step S40, or following a negative result in step S39, processing proceeds to step S41 in FIG. 10.

In step S41, the code assembler 39 decides whether the difference received from the black run count comparator 37 is greater than zero. If so, then in step S42, the code assembler 39 writes an add command and the absolute value of the difference in the code memory 40, followed by the run information for the current scanning line, and proceeds to step S46. The add command indicates that the difference was positive.

If the difference is not greater than zero, then in step S43, the code assembler 39 decides whether the difference is less than zero. If so, then in step S44, the code assembler 39 writes a subtract command and the absolute value of the difference in the code memory 40, followed by the run information for the current scanning line. The subtract command indicates that the difference was negative.

If the difference is not negative, that is, if the difference is equal to zero, then in step S45, the code assembler 39 writes the run information for the current scanning line in the code memory 40. Step S44 or step S45 is followed by step S46.

In step S46, the code assembler 39 decides whether there is another scanning line to be coded, and returns to step S31 if there is. If no next scanning line is present, then in step S47, the offset value maintained in the offset counter 38 is tested. If the offset value is zero, the coding process ends. If the offset value is not zero, the code assembler 39 writes an offset command and the offset value in the code memory 40; then the coding process ends.

In steps S42, S44, and S45, the code assembler 39 preferably writes the run lengths of black runs, and the absolute difference values following the add and subtract commands, in a variable-length coded form, using short code words for small values.

FIG. 11 shows the configuration of a decoder for decoding the coded image data produced by the coder in FIG. 8. The decoder comprises a code memory 41, a decoding section 42 having a command analyzer 43 and black run counter 44, and an image memory 45.

The code memory 41 is similar to the code memory 21 in the first embodiment. The decoding section 42 reads the coded image data stored in the code memory 41 in sequence from beginning to end, generates decoded image data, and writes the decoded image data in the image memory 45. The command

analyzer 43 analyzes commands occurring in the coded image data. The black run counter 44 keeps track of the black run count. The image memory 45 stores the decoded image data pending printing or other use.

The operation of the decoder in FIG. 11 will be described with reference to FIG. 12.

In step S51, the decoding section 42 reads data from the code memory 41. In step S52, the command analyzer 43 decides whether an offset command occurs in the data read in step S51. If so, in step S53 the decoding section 22 generates the number of white lines specified by the offset value following the offset command, and writes these lines into the image memory 45.

Following step S53, the decoding section 42 resets the black run counter 44 to zero in step S54, then proceeds to step S62 (described below).

If there is no offset command in the data read in step S51, then in step S55, the command analyzer 43 decides whether an add command is present. If so, then the decoding section 42 decodes the absolute difference value following the add command in step S55, and adds this value to the count in the black run counter 44 in step S57.

Following step S57, in step S58 the decoding section 42 reads and decodes further data from the code memory 41 as necessary to obtain run information for one scanning line, generates a decoded scanning line containing black runs of the specified lengths in the specified starting positions, and writes this scanning line in the image memory 45. The amount of run information to be read is determined from the count given by the black run counter 44. Processing next proceeds to step S62 (described below).

If the command analyzer 43 does not find an add command in step S55, then in step S59, the command analyzer 43 decides whether there is a subtract command in the data read in step S51. If so, then the decoding section 42 decodes the absolute difference value following the subtract command in step S60, subtracts this value from the count in the black run counter 44 in step S61, and proceeds to step S58 (described above) to decode one scanning line. If no subtract command is present, processing proceeds directly from step S59 to step S58.

Following step S54 or S58, in step S62 the decoding section 42 decides whether the code memory 41 contains any more data to be decoded. If so, processing returns to step S51 to begin decoding the next scanning line. If not, the decoding process ends.

If the image coded in the second embodiment comprises mostly background area, then the length of the black runs tends to be short. Moreover, in most images, nearby pixels tend to be strongly correlated, so when the black run count changes, the change tends to be small. By using a variable-length code that codes small values in a small number of bits, the second embodiment can reduce the average number of bits needed to code run lengths in the run information, and the average number of bits needed to code the absolute difference values following add and subtract commands, thereby reducing the size of the coded data.

Like the first embodiment, the second embodiment is computationally simple, does not require the storage of large amounts of temporary data, and enables images to be coded and decoded at high speed with only modest memory usage.

The present invention can also be used to code color images. An example is an image comprising pixels of the colors black, white, blue, and red, the white pixels being background pixels. It will be assumed below that there is no gray scale, so each pixel has one of only four possible values, corresponding to the colors black, white, blue, and red.

A run in this type of image refers to a run of consecutive pixels having the same pixel value or color. For example, a blue run is a consecutive run of blue pixels.

As a third embodiment of the invention, FIG. 13 shows a coder and FIG. 14 shows a decoder suitable for this type of image.

The coder in FIG. 13 is quite similar to the coder in the first embodiment, shown in FIG. 2. The image memory 51 stores image data in which the pixel values are, for example, two-bit values. The run analyzer



52 reads one scanning line at a time. The run position analyzer 53 analyzes runs of each color separately, and obtains run information giving the position and color of each black run, blue run, and red run. The run counter 54 determines the total number of runs of these three colors in each scanning line.

The run information buffer 55 stores the run information obtained by the run position analyzer 53. For each run, the run information comprises the starting x-coordinate, the ending x-coordinate, and the color (black, blue, or red) of the run.

The run count comparator 56 and offset counter 57 receive the total run count obtained by the run counter 54, representing the combined number of black, blue, and red runs in one scanning line, and operate in the same way as the black run count comparator 16 and offset counter 17 in the first embodiment. The run count comparator 56 compares the total run counts of the current scanning line and the preceding scanning line, and produces a match signal or an unmatched signal. The offset counter 57 maintains an internal offset value representing the number of consecutive white lines, and outputs the offset value whenever the total run count changes from zero to a non-zero value.

The code assembler 58 receives the total run count from the run counter 54, the match and unmatched signals from the run count comparator 56, and the offset values output by the offset counter 57, reads run information from the run information buffer 55, and writes offset commands, run commands, and run information in the code memory 59. The only difference from the coded data produced in the first embodiment is that the run information for each run now includes a pixel color value as well as the starting and ending positions.

The decoder shown in FIG. 14 has a code memory 61, a decoding section 62, a command analyzer 63, and an image memory 64. These elements operate in the same way as the corresponding elements in FIG. 5 in the first embodiment, except that the decoding section 62 reads color information as part of the run information, and generates scanning lines with black, blue, and red runs.

The operation of the coder and decoder in the third embodiment follows the flowcharts shown in FIGS. 3, 4, and 6 for the first embodiment, so a repeated description will be omitted.

For sparse images with a restricted number of colors, the third embodiment provides the advantages noted in the first embodiment: fast coding and decoding, small memory requirements, and a high compression ratio. The third embodiment is also useful for coding sparse gray-scale images having a comparatively small number of gray levels, a run in this case being a run of pixels with identical gray levels.

FIGS. 15 and 16 illustrate a coder and decoder according to a fourth embodiment, which encodes gray-scale information in a different way. The gray scale considered in the fourth embodiment has, for example, a minimum level corresponding to black, a maximum level corresponding to white, and at least one intermediate level. White pixels are considered to be background pixels.

The fourth embodiment counts runs of non-white pixels. The pixels in a run need not have the same gray level; the only requirement is that none of the pixels have the maximum (white) level. The run information describing a run of nonwhite pixels comprises the x-coordinates of the starting and ending pixels, the gray level of the starting pixel, also referred to below as the starting gray level, and the gray level of the ending pixel, also referred to as the ending gray level.

The coder in FIG. 15 comprises an image memory 71 that stores a gray-scale image, which is read one scanning line at a time by the run analyzer 72. The run position analyzer 73 analyzes runs of non-white pixels, and obtains run information giving the x-coordinates and gray levels of the starting and ending pixels in each run. The run counter 74 counts the total number of non-white runs in each scanning line. The run information buffer 75 temporarily stores the run information obtained by the run position analyzer 73 for each scanning line.

The run count comparator 76 and offset counter 77 receive the run count obtained by the run counter 74, and operate as did the black run count comparator 16 and offset counter 17 in the first embodiment. The run count comparator 76 compares the run count of each scanning line with the run count of the preceding scanning line, and produces a match signal or an unmatched signal. The offset counter 77 maintains an internal offset value representing the number of consecutive white lines, and outputs the offset value whenever the run count changes from zero to a non-zero value.

The code assembler 78 writes offset commands, run commands, and run information in the image memory 79, using information obtained from the run counter 74, the run count comparator 76, the offset counter 77, and the run information buffer 75.

The decoder shown in FIG. 16 has a code memory 81, a decoding section 82 with a command analyzer 83, an interpolator 84, and an image memory 85. The command analyzer 83 operates in the same way as the command analyzer in the first embodiment.

The decoding section 82 reads offset commands, run commands, and run information from the code memory 81, generates white lines, and stores the most recent run count as in the first embodiment. Run information read by the decoding section 82 is sent to the interpolator 84. The interpolator 84 uses the starting and ending x-coordinates to calculate the length of each run, subtracts the starting gray level from the ending gray level to obtain a difference value, and divides this difference value by the run length minus one to obtain a gray-level increment value. The interpolator 84 then assigns the starting gray level to the first pixel in the run, and assigns a gray level to each other pixel in the run by adding the gray-level increment value to the gray level of the preceding pixel. The decoding section 82 rounds these gray levels off as necessary, and writes the resulting pixel values in the image memory 85, inserting white pixels between runs, to generate decoded scanning lines.

Except for the interpolation step, the operation of the coder and decoder in the fourth embodiment follows the flowcharts shown in FIGS. 3, 4, and 6, so further description will be omitted.

For sparse gray-scale images, the fourth embodiment provides the same advantages as in the preceding embodiments: fast coding and decoding, small memory requirements, and a high compression ratio. Because the decoder interpolates gray levels in runs, the decoded image is not in general identical to the original image, but for certain types of images the differences are non-existent or negligible. The fourth embodiment is particularly useful in coding sparse three-dimensional images in which gray levels are used to express the depth dimension.

The invention can be practiced by embodying the invented coder and decoder by means of arithmetic, logic, and memory circuits in, for example, one or more semiconductor integrated circuits. The invention can also be practiced by embodying the functions of the coder and decoder in programs executable by a general-purpose processor, such as a microprocessor used in a computer or printer. The programs can be supplied at low cost on machine-readable media, such as rotating disk media or non-volatile memory media, or through communication networks.

As a variation of the fourth embodiment, the run analyzer 72 in the coder can compute the run length and gray-level increment value of each run, and store these values, together with the position and gray level of the starting pixel, as the run information in the run information buffer 75. The decoding process can thereby be simplified and speeded up.

As a variation of the third embodiment, the coder can store the starting position, run length, and pixel value of each run, and employ add and subtract commands instead of run commands, as in the second embodiment.

As a variation of the second embodiment, segments with non-zero run counts can be indicated by an add command, followed by the signed algebraic value of the difference from the preceding run count, instead of an add or subtract command followed by the absolute difference.

Those skilled in the art will recognize that further variations are possible within the scope claimed below.

Data supplied from the *esp@cenet* database - Worldwide

## IMAGE CODING/DECODING METHOD AND ITS SYSTEM

Claims of correspondent: **US6301300**

What is claimed is:

1. A method of coding an image in which pixels are organized into successive scanning lines and take on a plurality of values, pixels having a particular one of said plurality of values being background pixels, comprising the steps of:  
counting runs of non-background pixels in each scanning line, thereby obtaining a run count for each said scanning line;  
counting consecutive scanning lines having run counts of zero;  
coding a segment of consecutive scanning lines having identical run counts of zero as a first command and a first value, the first value indicating the number of scanning lines in the segment; and  
coding a segment of scanning lines having identical run counts not equal to zero as a second command and a second value, the second value indicating the run count of each scanning line in the segment, followed by one-dimensional run information describing each run of non-background pixels in each scanning line in the segment.
2. The method of claim 1, wherein said second value specifies the run count of the scanning lines in the segment coded by said second command.
3. The method of claim 1, wherein said second value specifies a difference from a run count of a preceding segment.
4. The method of claim 3, wherein said difference is an absolute difference, and said second command is selected from an add command and a subtract command.
5. The method of claim 1, wherein the run information for each run of non-background pixels comprises a starting coordinate and an ending coordinate.
6. The method of claim 1, wherein the run information for each run of non-background pixels comprises a starting coordinate and a run length.
7. The method of claim 1, wherein the run information for each run of non-background pixels comprises a single pixel value shared by all of the pixels in the run.
8. The method of claim 1, wherein the run information for each run of non-background pixels comprises a starting gray level and an ending gray level.
9. The method of claim 1, wherein the run information for each run of non-background pixels comprises a starting gray level and a gray-level increment value.
10. A machine-readable recording medium storing a machine-executable program for coding an image by the method of claim 1.
11. A method of decoding image data coded by the method of claim 1, comprising the steps of:  
analyzing commands in said image data;  
decoding each said first command and first value by generating scanning lines comprising only background pixels;  
decoding each said second command and second value, thus determining an internal run count; and  
decoding the run information following each said second command and second value by generating scanning lines having runs of non-background pixels as described by the run information, placing a number of runs of non-background pixels equal to said internal run count in each scanning line thus generated.
12. The method of claim 11, further comprising the steps of:  
setting said internal run count to zero when each said first command is detected; and

modifying said internal run count by an arithmetic operation, using said second value, when each said second command is detected.

13. The method of claim 11, further comprising the step of assigning gray levels to the pixels in each run of non-background pixels by interpolating between a starting gray level and an ending gray level, the starting gray level and the ending gray level being given in the run information.

14. The method of claim 11, further comprising the step of assigning gray levels to the pixels in each run of non-background pixels by successively adding a gray-level increment value given in the run information to a starting gray level given in the run information.

15. A machine-readable recording medium storing a machine-executable program for decoding coded image data by the method of claim 11.

16. A coder for coding an image in which pixels are organized into successive scanning lines and take on a plurality of values, pixels having a particular one of said plurality of values being background pixels, comprising:

a run analyzer for counting runs of non-background pixels in each scanning line in said image, thereby obtaining a run count for each said scanning line, and obtaining one-dimensional run information describing each of said runs of non-background pixels;

a run information buffer coupled to said run analyzer, for temporarily storing said run information for each said scanning line;

a run count comparator coupled to said run analyzer, for comparing the run count of each said scanning line with the run count of a preceding scanning line;

an offset counter coupled to said run analyzer, for counting consecutive scanning lines having run counts of zero; and

a code assembler coupled to said run count comparator, for outputting a first command and first value when said run count changes from zero to a non-zero value, said first value being provided by said offset counter, outputting a second command and second value when said run count changes from any value to a different non-zero value, said second value indicating said different non-zero value, and outputting the run information stored in said run information buffer.

17. The coder of claim 16, wherein said run count comparator sends said code assembler a signal indicating whether the run count of each said scanning line matches the run count of the preceding scanning line.

18. The coder of claim 17, wherein said second value is the run count obtained by said run analyzer.

19. The coder of claim 16, wherein said run count comparator sends said code assembler a difference value between the run count of each said scanning line and the run count of the preceding scanning line.

20. The coder of claim 19, wherein said second value is said difference value.

21. The coder of claim 19, wherein said second value is an absolute value of said difference value, and said second command indicates whether said difference value is positive or negative.

22. The coder of claim 16, wherein said run information comprises a starting coordinate and an ending coordinate for each run of non-background pixels.

23. The coder of claim 16, wherein said run information comprises a starting coordinate and a run length for each run of non-background pixels.

24. The coder of claim 16, wherein said run information comprises a single pixel value for each run of non-background pixels, said single pixel value being shared by all of the pixels in the run.

25. The coder of claim 16, wherein said run information comprises a starting gray level and an ending gray level for each run of non-background pixels.

26. The coder of claim 16, wherein said run information comprises a starting gray level and a gray-level increment value for each run of non-background pixels.

27. The coder of claim 16, further comprising an image memory storing said image in bit-mapped form.

28. The coder of claim 16, further comprising an image memory storing said image in run-length coded form.

29. A decoder for decoding coded image data describing an image in which pixels are organized into successive scanning lines and take on a plurality of values, pixels having a particular one of said plurality of values being background pixels, comprising:

a command analyzer for analyzing said coded image data to detect a first command designating a segment of consecutive scanning lines having only said background pixels, and a second command designating a segment of consecutive scanning lines having identical numbers of runs of non-background pixels; and

a decoding section coupled to said command analyzer, for reading, from said coded image data, a first value accompanying said first command, a second value accompanying said second command, and one-dimensional run information describing runs of non-background pixels, generating consecutive scanning lines of background pixels in quantities specified by said first value, and generating scanning lines having the runs described by said run information, determining from said second value how many runs to place in each scanning line.

30. The decoder of claim 29, further comprising a run counter which is reset to zero when said first command is detected, and is incremented and decremented according to said second command and said second value.

31. The decoder of claim 29, further comprising an interpolator for assigning gray levels to the pixels in each run of non-background pixels by interpolating between a starting gray level and an ending gray level, the starting gray level and the ending gray level being given in said run information.

32. The decoder of claim 29, further comprising an interpolator for assigning gray levels to the pixels in each run of non-background pixels by successively adding a gray-level increment value to a starting gray level, the gray-level increment value and starting gray level being given in said run information.

33. An image coding and decoding apparatus for coding and decoding an image in which pixels are organized into successive scanning lines and take on a plurality of values, pixels having a particular one of said plurality of values being background pixels, comprising:

a run analyzer for counting runs of non-background pixels in each scanning line in said image, thereby obtaining a run count for each said scanning line, and obtaining one-dimensional run information describing each of said runs of non-background pixels;

a run information buffer coupled to said run analyzer, for temporarily storing said run information for each said scanning line;

a run count comparator coupled to said run analyzer, for comparing the run count of each said scanning line with the run count of a preceding scanning line;

an offset counter coupled to said run analyzer, for counting consecutive scanning lines having run counts of zero;

a code assembler coupled to said run count comparator, for outputting a first command and first value when said run count changes from zero to a non-zero value, said first value being provided by said offset counter, outputting a second command and second value when said run count changes from any value to a different non-zero value, said second value indicating said different non-zero value, and outputting the run information stored in said run information buffer, thereby generating coded image data;

a code memory coupled to said code assembler, for storing said coded image data;

a command analyzer coupled to said code memory, for analyzing said coded image data to detect said first command and said second command; and

a decoding section coupled to said command analyzer, for reading said first value, said second value, and said run information from said coded image data, generating consecutive scanning lines of background pixels according to said first command and said first value, and generating scanning lines having the runs described by said run information, determining from said second value how many runs to place in each scanning line.

34. The image coding and decoding apparatus of claim 33, wherein said run count comparator sends said code assembler a signal indicating whether the run count of each said scanning line matches the run count of the preceding scanning line.

35. The image coding and decoding apparatus of claim 34, wherein said second value is the run count obtained by said run analyzer.
36. The image coding and decoding apparatus of claim 33, wherein said run count comparator sends said code assembler a difference value between the run count of each said scanning line and the run count of the preceding scanning line.
37. The coder of claim 36, wherein said second value is said difference value.
38. The image coding and decoding apparatus of claim 37, further comprising a run counter coupled to said command analyzer, which is reset to zero when said first command is detected, and incremented and decremented according to said second command and said second value.
39. The image coding and decoding apparatus of claim 36, wherein said second value is an absolute value of said difference value, and said second command indicates whether said difference value is positive or negative.
40. The image coding and decoding apparatus of claim 39, further comprising a run counter coupled to said command analyzer, which is reset to zero when said first command is detected, and incremented and decremented according to said second command and said second value.
41. The image coding and decoding apparatus of claim 33, wherein said run information comprises a starting coordinate and an ending coordinate for each run of non-background pixels.
42. The image coding and decoding apparatus of claim 33, wherein said run information comprises a starting coordinate and a run length for each run of non-background pixels.
43. The image coding and decoding apparatus of claim 33, wherein said run information comprises a single pixel value for each run of non-background pixels, said single pixel value being shared by all of the pixels in the run.
44. The image coding and decoding apparatus of claim 33, wherein said run information comprises a starting gray level and an ending gray level for each run of non-background pixels.
45. The image coding and decoding apparatus of claim 44, further comprising an interpolator coupled to said decoding section, for assigning gray levels to the pixels in each run of non-background pixels by interpolating between said starting gray level and said ending gray level.
46. The image coding and decoding apparatus of claim 33, wherein said run information comprises a starting gray level and a gray-level increment value for each run of non-background pixels.
47. The image coding and decoding apparatus of claim 46, further comprising an interpolator coupled to said decoding section, for assigning gray levels to the pixels in each run of non-background pixels by successively adding said gray level increment value to said starting gray level.
48. The image coding and decoding apparatus of claim 33, further comprising an image memory storing said image in bit-mapped form.
49. The image coding and decoding apparatus of claim 33, further comprising an image memory storing said image in run-length coded form.

---

Data supplied from the **esp@cenet** database - Worldwide